electronics

MDPI

Article

Utilizing Latent Diffusion Model to Accelerate Sampling Speed and Enhance Text Generation Quality

Chenyang Li [1,2], Long Zhang [1,2,*] and Qiusheng Zheng [1,2]

[1] The Frontier Information Technology Research Institute, Zhongyuan University of Technology, Zhengzhou 450007, China; lcy@zut.edu.cn (C.L.); zqs@zut.edu.cn (Q.Z.)
[2] Henan Key Laboratory on Public Opinion Intelligent Analysis, Zhengzhou 450007, China
* Correspondence: zhanglong@zut.edu.cn

Abstract: Diffusion models have achieved tremendous success in modeling continuous data modalities, such as images, audio, and video, yet their application in discrete data domains (e.g., natural language) has been limited. Existing methods primarily represent discrete text in a continuous diffusion space, incurring significant computational overhead during training and resulting in slow sampling speeds. This paper introduces LaDiffuSeq, a latent diffusion-based text generation model incorporating an encoder–decoder structure. Specifically, it first employs a pretrained encoder to map sequences composed of attributes and corresponding text into a low-dimensional latent vector space. Then, without the guidance of a classifier, it performs the diffusion process for the sequence's corresponding latent space. Finally, a pretrained decoder is used to decode the newly generated latent vectors, producing target texts that are relevant to themes and possess multiple emotional granularities. Compared to the benchmark model, DiffuSeq, this model achieves BERTScore improvements of 0.105 and 0.009 on two public real-world datasets (ChnSentiCorp and a debate dataset), respectively; perplexity falls by 3.333 and 4.562; and it effectively quadruples the text generation sampling speed.

Keywords: diffusion model; sequence diffusion; pretrained models; prompt; text generation; controllable emotion generation; fine-grained emotion

Citation: Li, C.; Zhang, L.; Zheng, Q. Utilizing Latent Diffusion Model to Accelerate Sampling Speed and Enhance Text Generation Quality. *Electronics* **2024**, *13*, 1093. https://doi.org/10.3390/electronics13061093

Academic Editor: Cecilio Angulo

Received: 14 February 2024
Revised: 12 March 2024
Accepted: 15 March 2024
Published: 15 March 2024

1. Introduction

Artificial Intelligence Generated Content (AIGC) is a method that utilizes artificial intelligence technology to automatically create articles, audio, and video. In recent years, with the continuous development and application of artificial intelligence technology, an increasing number of institutions and companies have started to experiment with AIGC to generate a large amount of content quickly and at a low cost, thereby meeting the needs across different fields. Against this backdrop, utilizing user emotional characteristics and stance attributes to guide the generation process, with the goal of automatically generating social text that adapts to specific attributes, has become one of the hot research topics in text generation technology. It also holds a broad range of application prospects [1]. Additionally, generating large-scale category-attribute text can also significantly alleviate the difficulty of obtaining large-scale labeled training datasets.

Therefore, a good, controllable text generation system is crucial and can be used to generate directive texts for various complex social scenarios. Currently, there are primarily two types of controllable text generation: one is template-based automated generation, and the other is deep learning-based automated generation. In the context of deep learning, methods such as Seq2Seq (sequence-to-sequence) [2] and the attention mechanism [3] have been widely applied in text generation systems and have achieved commendable success. However, they still have some shortcomings, such as generating sentence structures that may not be smooth or emotions that are not rich enough. With the emergence of some large-scale pretrained models, such as BART (Bidirectional and Auto-Regressive Transformers) [4]

and GPT-2 (Generative Pretrained Transformer) [5], it has become possible to generate high-quality text content in bulk with a relatively low barrier to entry. However, the uncontrollability of these models limits their application scope. As illustrated in Figure 1, if we need some negative reviews about books and we use pretrained models like GPT-2, BART, or ChatGPT for generation, it can be seen that the results are fluent but do not meet our specific requirements. This failure arises because the generated texts need to fit into particular applications, requiring the narration of events, the expression of specific viewpoints and emotions, etc. Thus, these texts need to be not only coherent and fluent but also encompass specific content, stance, and emotional attributes. Controllable text generation models can exert attribute control over generated texts, broadening the potential application scope. The aim of controllable text generation is to produce texts with specific semantics. By generating texts of varying categories and emotions, machine-generated content can become more humanized [6,7].

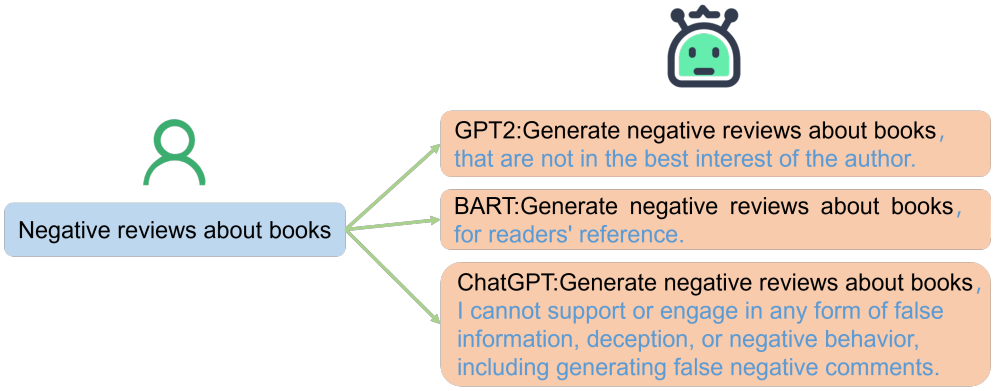

Figure 1. Controllable generation results of various models.

Addressing the aforementioned issues, we propose a novel diffusion sequence model for low-dimensional latent spaces called LaDiffuSeq. By treating control attributes and prompted text sequences [8] as sequences, the model employs a pretrained encoder to encode the sequences into a low-dimensional latent space and designs a sequence diffusion process without classifier guidance. This process fosters connections between sequences in the latent space, thereby achieving controllable generation [9,10]. Theme attributes can guide the content of the generated text, while emotional attributes can guide the emotional tone [11]. Experiments show that this model achieves high-quality text generation that is multi-attribute controllable while ensuring textual fluency. The main contributions of this paper are as follows:

1. Enriching the ChnSentiCorp dataset by adding topic attributes and expanding the binary emotion classification into seven finer-grained emotions.
2. Leveraging the capabilities of a pretrained decoder-encoder to encode text into a lower-dimensional latent vector space, solving the embedding conversion process from discrete text to a continuous space, and circumventing rounding loss problems inherent in traditional methods.
3. Designing a sequence diffusion process without classifier guidance, mapping the controllable text generation to a Seq2Seq task, and directly performing the diffusion process on low-dimensional vectors in latent space. This avoids the generation quality degradation problems associated with the introduction of classifiers.
4. Incorporating theme and emotion information, the generated texts become closer to the intended topics with fine-grained emotional expression.

2. Related Works

As a subfield of natural language processing, automatic text generation has advanced rapidly with the advent of pretrained language models, which are capable of generating

highly readable text. Language models that are improvements based on the transformer [12] architecture stand out in the field of text generation. For instance, pretrained models such as GPT-2 and BART, which are trained with extensive web text data, achieve excellent results in text generation due to their autoregressive properties. However, these general-purpose models have been challenging to apply in actual industry settings due to their lack of controllability, prompting many researchers to shift focus toward the technical study of controlled language models.

2.1. Text Generation Based on Pretrained Models

Keskar et al. [13] observed that although pretrained models like GPT-2 and BART possess the capability to generate high-quality text, the lack of integration with rules controlling the content of the generated text means that existing technologies struggle to automatically generate task-specific content. In response, they proposed the CTRL model, capable of controlled content generation. The core idea behind CTRL is the inclusion of control codes in the language model, classifying the data in the corpus, and appending a type descriptor before each specific sequence, thereby linking the corpus with its type. Dathathri et al. [14] introduced a plug-and-play model training framework known as Plug and Play Language Models (PPLM). This approach embeds one or multiple attribute classifiers within the language model training, guiding the output distribution of the language model to control textual attributes. This method significantly reduces dependency on data and hardware configurations for text generation tasks. However, PPLM still requires updating the parameters of large models, which results in slower inference speeds. As an improvement over PPLM, FUDGE (Controlled Text Generation With Future Discriminators) [15] does not update any parameters within the model but rather introduces a discriminator to predict whether the ongoing generated text conforms to the desired attributes. FUDGE theoretically decomposes conditional generation probabilities using Bayes' theorem, creates predictors by learning attributes of a portion of the sequence, and then uses the predictor's output to adjust the original language model's probability distribution, thus producing text with specific attributes.

2.2. Text Generation Based on Diffusion Models

Inspired by non-equilibrium thermodynamics, diffusion models introduce noise to the data distribution during the forward process and learn a reverse denoising process [16]. Song et al. [17] further applied this to high-quality image generation, and due to their iterative diffusion characteristics, they offer a more stable training and generation process, surpassing Generative Adversarial Networks (GANs) [18–20] in image generation. The denoising diffusion probabilistic model (DDPM) [21] has gained attention for having the ability to generate high-quality samples without adversarial training, and its quality of generation far exceeds that of other generative models. Song et al. [22] have realized faster sampling through the denoising diffusion implicit model. Successful image generation models like CLIP [23], Stable Diffusion [24], and Midijourney [25] have utilized these diffusion-based techniques [26].

The Gaussian noise addition process in diffusion models [27] mainly targets the continuous states of images [28] or waveforms [29], and clearly is not suitable for text tasks. To meet this challenge, in 2022, Jacob et al. [30] introduced the diffusion process into discrete variables, defined a series of transition matrices, and conducted diffusion directly on discrete texts. They used transition matrices to probabilistically convert a discrete word into a mask or leave it unchanged over different time steps, constructing a diffusion model for discrete texts. Although this approach didn't produce high-quality texts or enable controllable generation, it represented an attempt to apply the diffusion model to the field of text generation. To address issues with the non-differentiability of discrete text, Li et al. [31] proposed a non-autoregressive language model based on continuous diffusion models, named Diffusion-LM. The authors defined a word embedding method that unifies the discrete-to-continuous states in the diffusion process. By denoising a sequence of

Gaussian vectors into word vectors, they generated intermediate latent variables whose continuous and hierarchical nature allowed a simple gradient-based algorithm to perform complex and controllable generation tasks. This model achieved good results, especially in terms of text diversity, but it falls short in sentence fluency. DiffuSeq [32] used an end-to-end training method without needing to train an additional classifier to control the denoising process, avoiding degradation issues due to decoding strategies, and thus improving sentence diversity without sacrificing quality.

With the rise of large-scale pretrained language models, text generation technology has become more sophisticated. Thanks to the inherent autoregressive decoding advantages of transformer-based models [33], large pretrained models like those in the GPT series have become a new paradigm for text generation [13]. However, the generative capabilities of non-autoregressive decoding models should not be overlooked. Although current pretrained models are already capable of producing fluent text, and controlled text generation and increased text diversity can be achieved through methods such as PPLM and FUDGE, this is done at the expense of text fluency [34,35].

3. Materials and Methods

The problem addressed in this paper can be defined as follows: given a control attribute w^x and the real text w^y, train a language model such that when w^x is inputted, the language model can output a high-quality target text $w^{y\prime}$ that conforms to w^x. Therefore, the controlled text generation task can be formalized as:

$$p(w^{y\prime}\left|w^x\right.) \propto p\left(w^{y\prime}\right)\cdot p(w^x\left|w^{y\prime}\right.) \tag{1}$$

This involves sampling from the conditional distribution $p(w^{y\prime}\left|w^x\right.)$, where w^x represents the control attribute, $p\left(w^{y\prime}\right)$ represents the output target text. The optimization of this ensures fluency, and $p(w^x\left|w^{y\prime}\right.)$ is used to complete the attribute control process. Its optimization ensures effective control of the attributes while maintaining fluid output text.

In this chapter, we mainly introduce the proposed sequence-controllable generation model based on latent space diffusion. As shown in Figure 2, the control attribute and the real text after the prompt [36] are represented by w^x and w^y, respectively, which constitute the sequence represented by w. Initially, a pretrained encoder is employed to encode the sequence w, and the sequence of encoded latent space is represented by z. $z_0 - z_t$ represent the state of z at time steps 0 to t. During the forward process, only the z^y part is noised, and like the calculation method of the traditional diffusion models, the state of each time step can be obtained by computing $q\left(z_t^y\left|z_{t-1}^y\right.\right)$. In the reverse process, only the z^y part is denoised to ensure consistency during training and prediction phases, to avoid reducing the fluency of the generated text. Meanwhile, w^x acts as a prompt to guide each step of the denoising process to ensure that each newly generated text conforms to z^x. At this point, the calculation method for each time step state is no longer the traditional diffusion $p_\theta\left(z_{t-1}^y\left|z_t^y\right.\right)$ but has changes to $p_\theta\left(z_{t-1}^y\left|z_t^y, z^x\right.\right)$. z^x serves as a latent vector representation of the control attribute and does not directly participate in the noising and denoising processes of the diffusion model, hence z^x remains unchanged at each time step. During the forward noising process, the objective is to establish a connection between the two different feature space vectors z^x and z^y in order to model the feature relation between the latent vector of the control attribute z^x and the latent vector of the text z^y. In the reverse denoising process, the main role of z^x is to act as a prompt to guide the diffusion model for conditional generation. Ultimately, the diffusion model can generate the latent vector $z^{y\prime}$ of target texts that conform to the control attribute latent vector z^x, which compose the new sequence of latent space represented by z'. Finally, a pretrained decoder combines the sequence in the latent space with attention information to decode it into a new text, represented as w'.

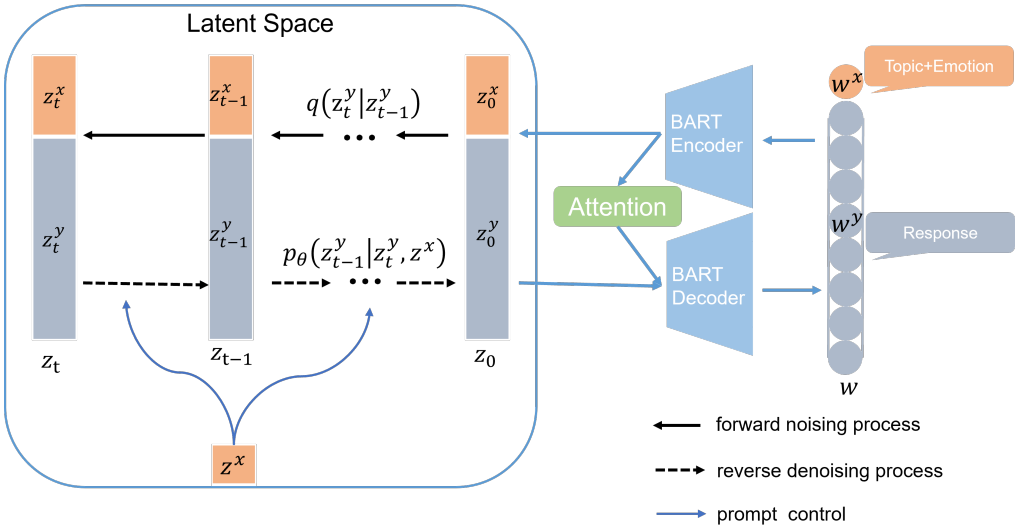

Figure 2. Sequence-controllable generation models with latent space diffusion.

3.1. Using BART for Text Sequence Encoding and Decoding

During the diffusion process, since text is discrete, it cannot be directly noised with Gaussian noise. Currently, there are two solutions: the first is to map the discrete text into a continuous representation space and apply noise as if it were an image, and the second is to generalize the diffusion process to text by introducing noise that is not just Gaussian noise but also includes insertion, editing, and deletion operations on the discrete text, and consider such processes as noising and denoising within the diffusion model [37]. As shown in Figure 3, we adopted a continuous diffusion model similar to the first method, but unlike previous methods, we attempted to use a pretrained model for encoding, mapping the input sequence into a continuous, latent, low-dimensional vector space. Since current sentence embeddings are mainly trained through contrastive learning (like Reimers and Gurevych) [38] rather than reconstruction, this makes sentence reconstruction challenging [39]. In simple terms, in the learned embedding vector space, similar sentences are distributed close to each other. We choose to use an encoder–decoder model because it can retain key information during the encoding process. Specifically, the pretrained encoder–decoder method encapsulates key information in the final hidden state of the encoder. We can then compress this final hidden state to create our embedding [40,41].

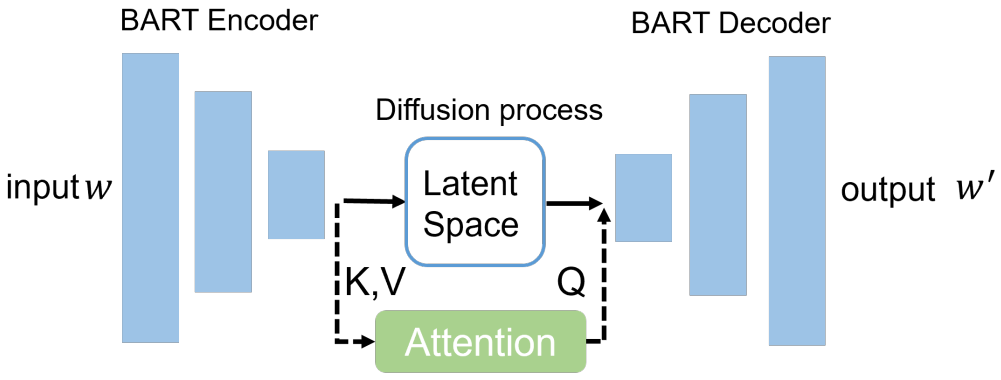

Figure 3. The encoding and decoding process of BART.

In theory, any transformer architecture that employs a pretrained encoder–decoder, such as Bart or T5, can be utilized, requiring only appropriate code adjustments to be adapted into our framework. However, upon examining in detail the parameters of the bart-base-Chinese and T5-base-Chinese models, it was discovered that the latter had almost

double the parameters of the former. Further online studies also showed that, compared to the T5-base-chinese model, the bart-base-Chinese model pretrained by Fudan University's Natural Language Processing team demonstrated a slight advantage in handling various kinds of Chinese tasks. This may be due to the use of high-quality datasets during the pretraining process. Therefore, choosing bart-base-Chinese as part of our model is a superior decision. For simplification, any mentions of BART hereafter refer to bart-base-Chinese. BART's vocabulary size is about 50k, indicating that the input vector dimension after one-hot encoding is 50k, which can be transformed into a 768-dimensional low-dimensional vector by the encoder. Due to the introduction of the attention mechanism, the encoding process not only retains the complete text feature information but also obtains attention values. These attention values indicate the degree of attention each word has to other words. During encoding and decoding, the attention mechanism typically connects the output of the encoder with the hidden states of the decoder, i.e., the keys (K) and values (V) in the encoder's attention values interact with the queries (Q) in the decoder, forming new attention values to ensure that the vector of the text latent space can be correctly decoded.

BART is a transformer that has both bidirectional language modeling and an autoregressive mechanism. Architecturally, BART adopts the standard transformer configuration with a 6-layer encoder and a 6-layer decoder structure. In our method, we freeze the encoder and decoder parameters of the pretrained BART model, using it solely to perform the encoding and decoding of sequences, while the actual training is dedicated exclusively to the diffusion process. Given a sequence composed of control attributes and corresponding texts, the control attributes and texts are separated by [SEP] and can be represented as one-hot vectors in the vocabulary. BART's encoding process includes three main components: word embeddings, the self-attention mechanism, and the feed-forward neural network. Firstly, the input words are converted into word embeddings with positional encoding. Then, a multi-head attention mechanism is utilized to compute the associations among the words in the sequence to better model the dependencies within. After the attention mechanism, the hidden representation at each position is used as input and processed through a fully connected-layer neural network to linearly transform and extract features from the hidden representation. The encoder is able to map the sequence into a continuous, low-dimensional latent vector space while retaining the information of the original sequence. At this point, the diffusion process acts like a black box, and after undergoing the diffusion process, the generated new vectors can be decoded back into new sequences by BART's decoder. The decoder's process is similar to the encoder's, producing a probability distribution for each word through the softmax function after processing by the feed-forward neural network.

For the BART model, the transformer architecture is commonly used for encoding and decoding, with the specific computation formulas as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

$$h_t = \text{Decoder}(y_{t-1}, \text{Encoder}(x)) \tag{3}$$

$$P(y_t|y_{t-1}, x) = \text{softmax}\left(h_t W^V\right) \tag{4}$$

Formula (2) is the core calculation mechanism of the attention operation, where Q, K, and V represent the matrix representations of queries, keys, and values, respectively. By taking the dot product of queries with keys, dividing by $\sqrt{d_k}$, and applying the softmax function, we obtain the attention weights for each query over all keys. These weights are then used to perform a weighted summation over the values, resulting in the final attention output. This attention mechanism is employed in both the encoder and the decoder components. $\text{Encoder}(x)$ indicates the encoder of the BART model that transforms the input sequence x into a contextual vector representation. $\text{Decoder}(y_{t-1}, \text{Encoder}(x))$ denotes the input to the decoder, which includes the first $t-1$ words of the generated

sequence y and the encoded context vector representation. h_t represents the hidden state of the decoder at time step t, and W^V is the output weight matrix of the decoder. The hidden state h_t of the decoder at time t is calculated from the previous time step's hidden state h_{t-1} and the output of the previous time step y_{t-1} (the word generated at the previous time step). Eventually, by computing the probability of the generated sequence y, the sequence with the highest probability is selected as the model's output.

Since we do not train an embedding model end-to-end, like Diffusion-LM, to map discrete text to a continuous representation space but instead utilize the encoding capabilities of a pretrained model to encode discrete text into a low-dimensional latent vector space, we avoid additional operations for gradient calculations on the generated distribution. Moreover, using a pretrained model as the initialization model allows the model to converge more rapidly.

3.2. Sequence Diffusion Process in Latent Space

The current Diffusion-LM first trains a language model capable of unconditional generation using a diffusion model and then separately trains attribute classifiers to guide the language model's output. While this approach enhances output diversity, it undoubtedly reduces the quality of language model generation. Therefore, we designed a latent sequence diffusion process without classifier guidance. Under the condition of guided control attributes, we directly diffuse the low-dimensional vectors in the latent space to achieve controlled generation of these low-dimensional vectors.

During the forward noise-adding process, from time step t_0 onwards, random noise is randomly applied to parts of the vectors in the sequence at each time step. Similar to the traditional diffusion model's noise-adding process, the amount of random noise added increases progressively with each time step, resulting in the sequence being in a state of complete Gaussian noise at the final moment. In the reverse denoising process, the vectors with added noise are gradually denoised. As illustrated in Figure 2, for the control attributes and text in the dataset, they are considered a sequence and encoded into a low-dimensional latent vector sequence. During forward diffusion, unlike traditional diffusion models that add noise to the entire latent vector sequence, the latent vectors corresponding to the control attributes part remain unchanged. Starting from z_0, noise is gradually added only to the latent vectors corresponding to the textual part, until the noise is advanced to the target space z_t^y, at which point z_t^y is in a state of complete Gaussian noise. In the reverse denoising process, the latent vectors corresponding to the control attributes part also remain unchanged, and only the latent vectors corresponding to the textual part are gradually denoised. The denoising at each step must conform to the control attribute's latent vector z^x. The reverse denoising computation process is changed from $p_\theta\left(z_{t-1}^y \middle| z_t^y\right)$ to $p_\theta\left(z_{t-1}^y \middle| z_t^y, z^x\right)$, indicating that the output at each moment is not only related to the output of the previous moment but also related to the control attribute's latent vector z^x. This is done with the objective that the spatial features of control attributes and text content can be combined, allowing them to establish a connection during the diffusion process, ultimately leading to the generation of new latent sequences controlled by the latent vectors of attributes z^x. This approach effectively treats text generation as a Seq2Seq task, training in pairs and essentially considering the control attributes as the input for text generation, akin to freezing the control attributes to predict the text [42].

During the diffusion process, controlling $z_{0:T}$ is equivalent to sampling from the posterior distribution in Equation (5):

$$p(z_{0:T}|z^x) = \prod_{t=1}^{T} p(z_{t-1}|z_t, z^x) \tag{5}$$

Specifically, at each step of diffusion:

$$p(z_{t-1}|z_t, z^x) \propto p(z_{t-1}|z_t) \cdot p(z^x|z_{t-1}, z_t) \tag{6}$$

The optimization objective at step t is:

$$\nabla x_{t-1} \log p(z_{t-1}|z_t, z^x) = \nabla z_{t-1} \log p(z_{t-1}|z_t) + \nabla z_{t-1} \log p(z^x|z_{t-1}) \quad (7)$$

The fluency regularization objective is:

$$\lambda \log p(z_{t-1}|z_t) + \log p(z^x|z_{t-1}) \quad (8)$$

The regularization optimization objective for attribute control is:

$$\lambda \log p(z_{t-1}|z_t, z^x) + \log p(z^x|z_{t-1}, z^x) \quad (9)$$

As shown in Equation (9), we combine the optimization of text fluency and control attributes into a single optimization objective, which serves as the basis for calculating gradients and optimizing parameters. By treating the control attributes and text as a sequence, we diffuse the sequence without classifier guidance to generate high-quality, emotion-controllable text. Compared to previous work, there is no need to train attribute classifiers separately, which ensures that the controllability of multiple attributes does not reduce the fluency of the text [22]. This also avoids the errors and training time associated with classifiers, making model training easier.

3.3. Context-Guided Strategy Based on Prompt

In the field of artificial intelligence, a prompt refers to a piece of text or instruction provided to the model to guide it to produce specific outputs. In text generation tasks, the application of the prompt method can help the model generate coherent and fluent text more effectively. Hence, when processing datasets, a prompt strategy is introduced. Upon loading the control attributes for the data, the process involves more than merely loading the attributes; it also includes appending prompt phrases before and after the control attributes to ensure that the concatenated sequence of control attributes and text is more fluid. For example, if the control attribute is "computer" (theme) and "like" (emotion), and the corresponding text content is "The appearance is stylish and atmospheric, and the price is acceptable. No overheating has been detected. Having used it for a few days, the quality seems good, and it's worth having." Then, after adding the prompt template, the control attribute becomes: "Favorable comment on computers." The final sequence content is: "Favorable comment on computers [SEP] The appearance is stylish and atmospheric, and the price is acceptable. No overheating has been detected. Having used it for a few days, the quality seems good, and it's worth having." When generating text, the model will attempt to interpret the prompt and produce a response based on that understanding. During model fine-tuning, it's the pretrained language models that accommodate various downstream tasks, which is evidenced by introducing auxiliary task losses into the pretrained model for fine-tuning to better adapt to downstream tasks. In this process, the pretrained model makes more compromises. However, a prompt represents the downstream tasks catering to the pretrained model. Specifically, this requires the reconstruction of different tasks to make them compatible with pretrained language models, which means that the downstream tasks are the ones making more sacrifices. A prompt can be seen as a way to retrieve knowledge already memorized in the pretrained language model. When a prompt is used to feed samples into the model, the model receives more cues during prediction; thus, it uses more information, which enables the language model to better understand the context and task requirements, leading to more accurate and fluent text generation. This is one of the advantages of pretrained models; the knowledge within a pretrained model can be mined with prompts, which may not be as readily achievable with models trained from scratch [23].

4. Experiment and Result Analysis

4.1. Dataset

Compared to English datasets, Chinese datasets pose more challenges in sentiment classification tasks. Firstly, Chinese is a very complex language, with rich and intricate semantics, grammar, and word structures. Unlike English, which has clear word boundaries, Chinese adds extra complexity to word segmentation. Furthermore, the polysemy phenomenon in Chinese is relatively prominent, adding additional difficulty to sentiment analysis. Therefore, this article chose the Chinese dataset for experimentation.

The target dataset for this controllable generation task is a review dataset containing multiple attributes, such as themes. However, current publicly available review datasets are based on sentiment classification tasks, which do not include theme attributes and only have two polarities of positive and negative, that is, only one attribute. These datasets are obviously not suitable for multi-attribute, controllable review generation. To validate the effectiveness of their method, the authors of this paper conducted experiments using two public Chinese datasets: the ChnSentiCorp review dataset and a debate dataset.

Fine-grained sentiment aims to categorize sentiment in text in a more nuanced and detailed manner. As opposed to traditional sentiment analysis (positive, negative), fine-grained sentiment analysis partitions sentiment into a broader range of categories; for instance, negative sentiment can be further refined into categories such as anger and sadness, etc., to capture the slight differences in sentiment within a text more accurately. The ChnSentiCorp dataset was introduced around 2014, when the research spotlight was on the polarity of positive and negative sentiment. As explorations into sentiment computation continue, the uncomplicated positive and negative sentiment categorization no longer meets the researchers' needs in sentiment computation studies, hence the emergence of the more detailed sentiment categorization approach we see today. However, there is currently a scarcity of publicly available Chinese sentiment datasets, and most researchers resort to using privately held data crawled from the web. This has led to ChnSentiCorp, a classic binary sentiment classification dataset, still being used to this day; moreover, this dataset's data quality is relatively high as it contains real examples. Therefore, based on the current state of research, using merely the positive and negative sentiment of the original ChnSentiCorp dataset for experimentation is far from sufficient. In our analysis of the ChnSentiCorp dataset, we discovered that each text within the dataset contains more detailed sentiment, and it may be misleading to assume that because the dataset only had positive and negative labels upon release, they only fall into the two polarities of positive and negative, when in fact, the positive and negative sentiment were only the research spotlight at the time of the dataset's release. Simultaneously, we found that the dataset comprises three themes: hotels, books, and computers, none of which were notated. Annotating this dataset with sentiment and theme attributes makes it align more closely with our task of multi-attribute controlled generation. In addition, expanding the binary classification into seven kinds of sentiments not only increases the difficulty of the experiment but also validates the robustness of the model, further proving whether our model can effectively generate text with even more controlled attributes.

For the sentiment classification task-based ChnSentiCorp dataset, it contains 12,000 review entries, each corresponding to a single positive or negative attribute. After removing duplicates and irrelevant data, a total of 9090 review entries were retained. The authors observed that this dataset covers three themes: Ctrip Hotel, Dangdang Book, and JD Computer. To perform multi-attribute expansion on this dataset, they first annotated the corresponding theme attributes, as shown in Table 1. To implement multi-granularity sentiment classification for this dataset, they first trained a seven-category fine-grained sentiment classification model using the OCEMOTION dataset, which has achieved a 99% accuracy rate on its test set and can predict seven emotions, including anger, disgust, sadness, fear, surprise, liking, and happiness. This analysis model was then used to predict sentiment granularity [43] for the ChnSentiCorp dataset, as shown in Table 2. Ultimately, the target dataset included three themes and seven sentimental control attributes. To verify

that the seven-category fine-grained sentiment classification model could also achieve high accuracy on the ChnSentiCorp dataset, the authors randomly selected 300 review entries for manual annotation and compared the results with the model predictions, achieving an accuracy rate of over 98%. It is worth noting that because the ChnSentiCorp dataset mainly includes product reviews, it is theoretically assumed that there will be few reviews related to the emotion of "fear." Therefore, in the subsequent model prediction, only 13 reviews were predicted to have the emotion of "fear," which is consistent with the actual data distribution.

Table 1. Distribution of topic attributes in dataset A.

Computer	Hotel	Books	Total
2741	3910	2773	9090

Table 2. Distribution of fine-grained sentiment attributes in dataset A.

Anger	Disgust	Sadness	Fear	Surprise	Liking	Happiness	Total
824	749	2741	13	228	2435	2100	9090

The debate dataset [44] originates from dozens of famous Chinese-language debate competitions in recent years. The text for each single section and single-party statement of each competition was obtained through speech-to-text conversion and manual verification and was further annotated by annotators for proposition sentences and interactive proposition pairs. This dataset comprises 16 debate topics, each containing multiple statements from both the affirmative and negative sides, totaling 3716 data entries. Eventually, this dataset included 16 debate topics and two positional control attributes.

Although these two datasets only comprise a few thousand entries and are not considered large relative to text generation tasks, they offer the opportunity to conduct experiments on controllable review generation using multi-attribute datasets in different domains. The ChnSentiCorp dataset provides review data with two attributes: theme and emotion, enabling experiments on different themes and emotional attributes to verify the multi-controllable nature of the generation model. At the same time, the debate dataset provides debate topics and stance attributes, enabling experiments on review generation under different debate topics and stances to further verify the performance of the method in multi-attribute control. Therefore, these two datasets provide a wide range of experimental scenarios to test and evaluate the effectiveness and robustness of controllable review generation methods. To ensure naming consistency, the ChnSentiCorp dataset will be referred to as Dataset A and the debate dataset as Dataset B in subsequent discussions.

4.2. Evaluation Metrics

In the field of automatic text generation, unlike text classification, there are no systematic automatic evaluation metrics. Assessing the quality of a text generation task requires considering multiple aspects, including semantic accuracy, diversity, consistency, and fluency of language [45]. First, BLEU [46] metrics are used to evaluate semantic accuracy and text diversity. Both BLEU and self-BLEU scores range from 0 to 1. A higher BLEU score indicates that the generated text meets requirements more closely and can convey the correct meaning, representing semantic accuracy. Conversely, self-BLEU represents diversity; a lower self-BLEU score indicates higher diversity and more creativity in the generated text [47,48]. However, traditional machine evaluation metrics like BLEU-N are more suitable for machine translation and text summarization tasks. Therefore, a new text quality evaluation metric, BERTScore [49,50], is introduced. Compared to the traditional BLEU evaluation method, BERTScore can better assess the semantic similarity and grammatical correctness of the generated text. Since the above metrics rely on reference texts to calculate similarity, a measure known as perplexity (PPL) [51] is introduced. This evaluates

the fluency of the generated text using a language model, comparing the probabilities of the generated text to those produced by GPT-2. A lower PPL value implies a better fit to GPT-2's output, indicating that the generated text is more in line with language norms and easier for people to comprehend. However, the aforementioned metrics only assess if sentences are fluent. As the task at hand is controlled generation, it is essential to ensure that the generated sentences are not only fluent but also adhere to the given control attributes. Therefore, various attribute classifiers are trained to assess whether the generated text complies with the preset attributes. Accuracy is used as a quantitative measure of the degree of control.

4.3. Experimental Results Analysis

The D3PM employs a discrete diffusion approach, which treats the masking of words as a noise addition process during diffusion and the decoding of masks back to text as a denoising step in the diffusion model [52]. When conducting diffusion on discrete text, there are not as many opportunities to add noise to the discrete text as there are pixels in images. With fewer timesteps available for adding noise, the language model learns less, leading to less fluent text generation. Through experimentation, it has been found that at 32 diffusion steps, the proposed method exceeds the performance of D3PM at 512 timesteps, and with even more diffusion steps, it achieves better results.

Similar to D3PM, continuous diffusion methods like Diffusion-LM, DiffuSeq, and SeqDiffuSeq [53] first map discrete text onto continuous representation vectors, then treat these vectors as if they were images to be noised. However, these methods use end-to-end word embeddings, meaning that the embedding models are trained simultaneously with the diffusion models. In experiments, it has been found that these models need 2000 diffusion steps to converge. Since the proposed approach diffuses low-dimensional vectors in latent space, which retains all the textual feature information, the diffusion process for low-dimensional feature vectors involves orders of magnitude fewer diffusion steps compared to other continuous diffusion methods. Setting the number of diffusion steps to 512 is sufficient for the language models to converge, meaning that LaDiffuSeq is four times faster than other continuous diffusion models in terms of sampling speed. Intuitively, diffusing in the continuous space of discrete text, the more timesteps available for noise addition, the more the language model can learn, and the more fluent the generated text [54]. As shown in Experiment Table 3, it was found that LaDiffuSeq achieved far better results at 512 diffusion timesteps than the continuous diffusion model DiffuSeq at 2000 timesteps. This is because DiffuSeq involves rounding errors during the mapping of continuous space vectors to discrete text. These errors accumulate during the text generation process, leading to a degradation in the quality of the generated text. The present study, which uses BART as both the encoder and decoder, only requires training the diffusion model and avoids the concurrent training of the embedding model, therefore eliminating these errors. At 512 timesteps, it also demonstrates performance that can compete with GPT-2. This performance is attributed to the powerful encoding and decoding capabilities of the pretrained model BART.

To illustrate the advantages of diffusion without classifier guidance, the LaDiffuSeq method is compared to a baseline model that employs classifier-guided diffusion, specifically Diffusion-LM. As shown in Table 4, LaDiffuSeq outperforms Diffusion-LM in both text fluency and quality. This is because Diffusion-LM uses a controllable generation method similar to PPLM, which necessarily alters the probability output of the original language model when introducing classifier guidance. In the process of fitting the controllable attributes of the classifier, the language model selects words that are less fluent but align with classifier attributes for output. This results in text that meets the control conditions but has significantly reduced fluency. Since the sequence diffusion model in this study does not use a classifier for secondary guidance, it completely avoids this problem.

Table 3. PPL scores of various models at different time steps (↓ indicates that smaller values are better, while ↑ indicates that larger values are better).

Method	Text Embedding Method	Time Steps	Dataset A	Dataset B
			ppl↓	ppl↓
LaDiffuSeq	BART	32	223.57	145.93
		64	97.88	83.65
		256	43.525	44.59
		512	31.085	38.635
D3PM	none	512	225.15	152.75
Diffusion-LM	end-to-end	2000	196.164	130.145
DiffuSeq	end-to-end	2000	34.418	43.197
SeqDiffuSeq	end-to-end	2000	67.877	47.917
GPT-2	GPT	1	38.7	35.78

Table 4. Comparison of evaluation metrics for various models.

Method	Dataset A				Dataset B			
	bleu↑	self_bleu↓	BERTScore↑	ppl↓	bleu↑	self_bleu↓	BERTScore↑	ppl↓
Diffusion-LM	0.256	0.402	0.547	196.164	0.268	0.451	0.587	130.145
DiffuSeq	0.478	0.499	0.567	34.418	0.875	0.917	0.930	43.197
SeqDiffuSeq	0.476	0.571	0.589	67.877	0.501	0.627	0.711	47.917
LaDiffuSeq (no prompt)	0.493	0.481	0.670	33.332	0.882	0.485	0.932	39.733
LaDiffuSeq	0.501	0.476	0.672	31.085	0.899	0.473	0.939	38.635

Also shown in Table 4, aside from the self_bleu metric where Diffusion-LM performs better than this model, LaDiffuSeq outperforms other continuous diffusion models like DiffuSeq and SeqDiffuSeq in various indicators. Unlike other continuous diffusion models that use end-to-end embeddings to map discrete text to a continuous space and then add noise, this method uses BART to decode discrete text into a low-dimensional latent space. This avoids the conversion losses that continuous diffusion models suffer during the mapping process from discrete text to a continuous space or from a continuous space back to discrete text, thereby resulting in higher text generation quality than other continuous diffusion models. Typically, language models struggle to balance the quality and diversity of generated content—improving text quality often comes at the expense of diversity. To increase the diversity of text generation in this study, a Beam Search decoding strategy [55] is applied when decoding the sampled latent vectors, thus expanding the search space. Different from a greedy search strategy that only considers the currently optimal result, this approach considers the top-k optimal results, aiding in generating more diverse text content. The research model achieves not only high-quality text but also good diversity [56].

In order to verify the effectiveness of the prompt method, the study conducted comparative experiments with and without the addition of prompts. As shown in Table 4, it was found that the adoption of prompt strategies led to significant improvements in all evaluation metrics. This comparison experiment affirmed that prompting can enhance the model's performance and provide critical reference and guidance for further optimization of the model.

To validate whether the language model affects the quality of the generated text under controlled conditions, experiments comparing unconditional and conditional generations were conducted. As demonstrated by Tables 5 and 6, by comparing the quality of text

generated under various attribute condition combinations, it was found that there were only minor changes in the various evaluation metrics, which could be attributed to the randomness inherent in the language model's outputs. Despite the imposition of multiple control attributes, the output quality of the language model remained virtually unaffected. This indicates that the diffusion sequence model designed in this study has excellent text generation capabilities and is able to maintain high-quality outputs under controlled conditions. It also suggests that the model is not only suitable for unconditional generation but is better suited for diverse and controllable generation under multiple attribute control conditions.

Table 5. Comparison of evaluation metrics for various control attribute combinations in dataset A.

Control Attributes	bleu↑	self_bleu↓	BERTScore↑	ppl↓
unconditional	0.505	0.469	0.675	31.069
3 themes	0.503	0.469	0.677	31.372
2 emotions	0.502	0.472	0.677	31.493
7 emotions	0.499	0.478	0.676	31.037
3 themes+2 emotions	0.503	0.473	0.673	31.100
3 themes+7emotions	0.501	0.476	0.672	31.085

Table 6. Comparison of evaluation metrics for various control attribute combinations in dataset B.

Control Attributes	bleu↑	self_bleu↓	BERTScore↑	ppl↓
unconditional	0.892	0. 473	0. 939	38.352
debate topic	0.895	0. 472	0. 933	38.424
stance	0.899	0. 476	0. 931	38.550
debate topic stance	0.899	0.473	0.939	38.635

To verify whether the text generated using the ChnSentiCorp dataset aligns with the given themes and sentiments, an assessment of themes was first conducted. The language model produced 100 comments each for three different themes: hotels, computers, and books, which were then evaluated using a topic classifier. The results indicated that all 300 texts conformed to their corresponding themes. For the assessment of sentiment granularity, initially, it was only required to distinguish between positive and negative sentiments. The language model generated 100 positive and 100 negative reviews, respectively, and these texts were evaluated using a sentiment classifier. It was found that all 200 reviews corresponded to the appropriate sentiments. In order to avoid the language model potentially generating texts that can be easily classified by attribute classifiers, we further annotate these test data manually to verify the accuracy of the topic classifiers and sentiment classifier predictions. During the annotation process, we found that a few texts were hard to manually distinguish (for example, it's hard for an annotator to definitively determine whether the sentence "The price is a bit expensive, bought it for a friend" belongs to the computer or book theme). After excluding these texts that are hard to distinguish manually, the accuracy rate of manual annotation for the topic is 99%, and that for sentiment is 97%. Although there are minor differences between the tags marked manually and those predicted by the classifier, this difference is acceptable. This also fully demonstrates that our attribute classifiers can temporarily represent manual annotation for predictions. Through this verification, it further indicates that, under the controllable generation circumstances of three themes and two sentiments, the method in this paper can achieve an accuracy rate close to 100%.

Subsequently, positive and negative sentiments were further differentiated into seven more granular emotions. The language model created 100 comments for each emotion type, which were then assessed using a sentiment classifier. According to the results in Figure 4, the prediction accuracy for the emotions of sadness, anger, liking, happiness, and disgust was very high. However, the accuracy for fear and surprise was lower. The lower performance for these two emotions was because there were fewer instances of these emotions in the training set, resulting in the language model learning less about these specific emotions. Therefore, the accuracy of emotion generation is positively correlated with the number of instances of each emotion in the training set. These results further verify the characteristics of the distribution of sentiment categories in the ChnSentiCorp dataset; that is, certain emotional categories may occur less frequently. Despite fewer accurate predictions in the "fear" sentiment category, the model as described in the paper still performed well in predicting other sentiment categories. Additionally, the same method was used to evaluate arguments and stances in a debate dataset, and the end results were also consistent with their corresponding attributes.

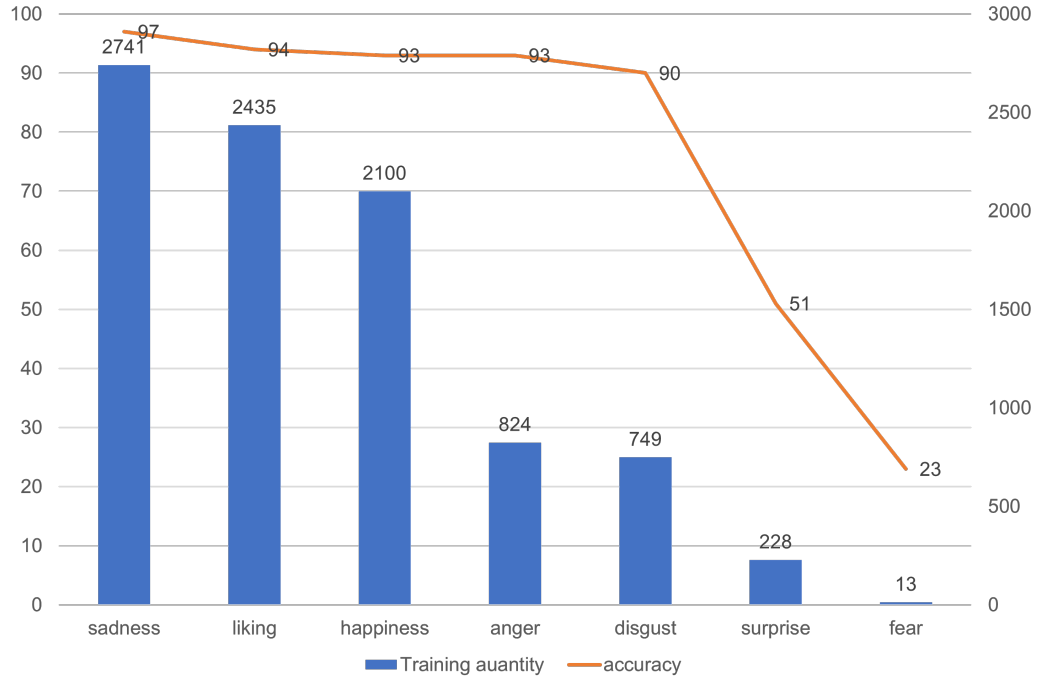

Figure 4. Number and predictive accuracy of each corresponding emotion.

As illustrated in Figures 5 and 6, by randomly selecting and displaying some of the generated results, it can be seen that the model is capable of generating target texts in two datasets through the control of two attribute dimensions. Additionally, the majority of the generated texts adhere to the target attributes, encompass more fine-grained emotions, have higher readability, and conform to the characteristics of real texts within the datasets.

theme	emotion	output	compliance
Hotel	happiness	Front desk service is very good, efficient, and the attitude is also very nice.	✓
	sadness	I highly doubt the evaluation standards of a five-star hotel; they seem to be worse than even those of a three-star hotel. I'm very disappointed!	✓
	fear	The breakfast food is quite average and doesn't match the level of service. I hope for further improvements.	×
Computer	liking	The appearance is nice, performance is excellent, heat dissipation is decent, the palm rest temperature is not high, and the overall usage feels quite comfortable.	✓
	anger	How can there be a price drop right after I bought it, It's really infuriating.	✓
Books	disgust	The content is too simple, repetitive, my son is one year and five months old, clearly not interested in this. I prefer the Tom series.	✓
	sadness	I don't know if this book has been abridged or heavily edited. If that's the case, then it's really regrettable.	✓

Figure 5. Showcase of controllable generation results in dataset A (In the figure, ✓ indicates compliance, while ✗ indicates non-compliance).

debate topic	stance	output	compliance
Being assimilated is more frightening than being excluded	opposition	Thank you, Chairman, and hello, everyone. Today, our side believes that being assimilated is more frightening. Individuals inevitably exist within a certain environment, so the premise of being excluded or assimilated is the difference between the individual and their living environment. We find that the reason anything is terrifying is that it makes us lose valuable things. Therefore, our criterion for judging what is more frightening today is between exclusion and assimilation, which of these results in us losing something more precious.	✓
	negative	And moreover, we have found that many times, being excluded can lead to numerous psychological disorders, and it can even overcome the most instinctive fear of death in individuals. Therefore, we believe that the negative impact of this is immeasurable. Throughout your argument, you haven't addressed one crucial point with our side, which is: fundamentally, what is lost through assimilation?	✓

Figure 6. Showcase of controllable generation results in dataset B.

5. Conclusions

In this paper, a more granular division of themes and sentiment attributes was conducted on the ChnSentiCorp dataset, covering three themes: hotels, computers, and books, along with seven emotions: anger, disgust, sadness, fear, surprise, liking, and happiness. Our experiments have demonstrated that the continuous diffusion model can effectively model within the latent space of a pretrained encoder–decoder language model. We have also designed a latent diffusion model structure without classifier guidance, which ensures the generation of target texts that are highly consistent with theme attributes, diverse, and clusterable. This method has significantly enhanced the text generation capabilities of pretrained language models, enabling them to generate high-quality and controllable texts from specific data distributions.

The approach described in this paper allows users to design personalized generation characteristics according to their business needs. The outstanding performance of this task stems not only from the application of the diffusion model but also from its effective integration with pretrained models. The entire model architecture is complete and interpretable, allowing for easy transfer to various fields within text generation, and holds promising practical applications. Future work will continue to explore the following aspects: in personalized review generation, incorporating more controllable information, such as personal preferences, intentions, etc. At the same time, we plan to extend our method to other generative tasks, such as text summarization and question generation. In terms of the model, due to the sequential structure of the diffusion model, which re-

sults in a prediction sampling speed several times slower than other generation models, Song et al. [22] proposed a denoising diffusion implicit model (DDIM) and redefined the sampling function to accelerate the generation process. Jolicoeur-Martineau et al. [57] designed a faster SDE solver for the reverse diffusion process, and in 2021, Salimans and Ho [58] refined a trained deterministic diffusion sampler into a new diffusion model that requires only half the sampling steps to generate a complete image. The latest research also introduced an orthogonal method called Denoising MCMC [59] to accelerate fraction-based sampling processes for diffusion models. Nevertheless, all the mentioned methods were designed for the field of computer vision, and methods for accelerating diffusion inference specifically for text generation remain to be explored. Evidently, diffusion models are more suitable for handling continuous data types such as videos, images, and audio. Due to the discrete nature of text data, this presents significant challenges for the application of diffusion models in natural language processing. This also explains why current large image generation models (such as Midijourney) opt for diffusion models, while large language models (like ChatGPT) do not employ this approach. Therefore, I believe that applying diffusion models to the field of natural language processing is still a novel method awaiting full exploration.

Author Contributions: Conceptualization, C.L. and L.Z.; methodology, C.L. and L.Z.; software, C.L.; validation, C.L.; formal analysis, C.L. and L.Z.; investigation, C.L.; resources, Q.Z.; data curation, C.L. and L.Z.; writing—original draft preparation, C.L.; writing—review and editing, L.Z.; visualization, C.L.; supervision, L.Z. and Q.Z.; project administration, L.Z. and Q.Z.; funding acquisition, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research Projects of Henan Higher Education Institutions, grant number 22B520054; the Songshan Laboratory Pre-research Project, grant number YYJC032022021; the Natural Science Foundation of Zhongyuan University of Technology, grant number K2023MS021.

Data Availability Statement: The source code in this study is available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
2. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1243–1252. [CrossRef]
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008. [CrossRef]
4. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1910.13461.
5. Fröhling, L.; Zubiaga, A. Feature-based detection of automated language models: Tackling GPT-2, GPT-3 and Grover. *PeerJ Comput. Sci.* **2021**, *7*, e443. [CrossRef]
6. Zhang, H.; Song, H.; Li, S.; Zhou, M.; Song, D. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Comput. Surv.* **2023**, *56*, 1–37. [CrossRef]
7. Sarzynska-Wawer, J.; Wawer, A.; Pawlak, A.; Szymanowska, J.; Stefaniak, I.; Jarkiewicz, M.; Okruszek, L. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Res.* **2021**, *304*, 114135. [CrossRef]
8. Yang, K.; Liu, D.; Lei, W.; Yang, B.; Xue, M.; Chen, B.; Xie, J. Tailor: A prompt-based approach to attribute-based controlled text generation. *arXiv* **2022**, arXiv:2204.13362.
9. Zhao, T.; Zhao, R.; Eskenazi, M. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv* **2017**, arXiv:1703.10960.
10. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv* **2022**, arXiv:2204.06125. [CrossRef]
11. Ghosh, S.; Chollet, M.; Laksana, E.; Morency, L.-P.; Scherer, S. Affect-lm: A neural language model for customizable affective text generation. *arXiv* **2017**, arXiv:1704.06851. [CrossRef]

12. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; pp. 38–45. [CrossRef]
13. Keskar, N.S.; McCann, B.; Varshney, L.R.; Xiong, C.; Socher, R. Ctrl: A conditional transformer language model for controllable generation. *arXiv* **2019**, arXiv:1909.05858.
14. Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; Liu, R. Plug and play language models: A simple approach to controlled text generation. *arXiv* **2019**, arXiv:1912.02164.
15. Yang, K.; Klein, D. FUDGE: Controlled text generation with future discriminators. *arXiv* **2021**, arXiv:2104.05218.
16. Sohl-Dickstein, J.; Weiss, E.A.; Maheswaranathan, N.; Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2256–2265.
17. Song, Y.; Ermon, S. Generative modeling by estimating gradients of the data distribution. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
18. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
19. Metz, L.; Poole, B.; Pfau, D.; Sohl-Dickstein, J. Unrolled generative adversarial networks. *arXiv* **2016**, arXiv:1611.02163.
20. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
21. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
22. Song, J.; Meng, C.; Ermon, S. Denoising diffusion implicit models. *arXiv* **2020**, arXiv:2010.02502.
23. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8748–8763.
24. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695. [CrossRef]
25. Oppenlaender, J. The creativity of text-to-image generation. In Proceedings of the 25th International Academic Mindtrek Conference, Tampere, Finland, 16–18 November 2022; pp. 192–202. [CrossRef]
26. Nichol, A.Q.; Dhariwal, P. Improved denoising diffusion probabilistic models. In Proceedings of the 2021 International Conference on Machine Learning, Virtual, 18–24 July 2021; ACM: New York, NY, USA, 2021; pp. 8162–8171.
27. Tashiro, Y.; Song, J.; Song, Y.; Ermon, S. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24804–24816.
28. Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv* **2021**, arXiv:2112.10741.
29. Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *arXiv* **2020**, arXiv:2009.09761.
30. Austin, J.; Johnson, D.D.; Ho, J.; Tarlow, D.; van den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 17981–17993.
31. Li, X.L.; Thickstun, J.; Gulrajani, I.; Liang, P.; Hashimoto, T.B. Diffusion-lm improves controllable text generation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 4328–4343. [CrossRef]
32. Gong, S.; Li, M.; Feng, J.; Wu, Z.; Kong, L. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv* **2022**, arXiv:2210.08933.
33. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
34. Lix, Q.; Wang, S.; Wang, Z.J.; Zhuj, W. Overview of natural language generation. *J. Comput. Appl.* **2021**, *41*, 1227–1235. [CrossRef]
35. Liu, X.-M.; Zhang, Z.-H.; Yang, C.-Y. Adversarial techniques for online social network text content. *J. Comput. Appl.* **2022**, *45*, 1571–1597. [CrossRef]
36. Li, J.; Tang, T.; Nie, J.-Y.; Wen, J.-R.; Zhao, X. Learning to transfer prompts for text generation. *arXiv* **2022**, arXiv:2205.01543.
37. Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; Yang, M.-H. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.* **2023**, *56*, 1–39. [CrossRef]
38. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
39. Zhao, L.; Zheng, K.; Zheng, Y.; Zhao, D.; Zhou, J. RLEG: Vision-language representation learning with diffusion-based embedding generation. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 42247–42258.
40. 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* **2022**, arXiv:2211.04236.
41. Gao, Z.; Guo, J.; Tan, X.; Zhu, Y.; Zhang, F.; Bian, J.; Xu, L. Difformer: Empowering diffusion model on embedding space for text generation. *arXiv* **2022**, arXiv:2212.09412.
42. Lin, Y.; Ji, H.; Liu, Z.; Sun, M. Denoising distantly supervised open-domain question answering. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1: Long Papers, pp. 1736–1745. [CrossRef]

43. Li, M.; Long, Y.; Lu, Q.; Li, W. Emotion corpus construction based on selection from hashtags. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; Springer: Cham, Switzerland, 2016; pp. 1845–1849.
44. Yuan, J.; Cheng, L.; He, R.; Li, Y.; Bing, L.; Wei, Z.; Liu, Q.; Shen, C.; Zhang, S.; Sun, C.; et al. Overview of argumentative text understanding for ai debater challenge. In Proceedings of the 2021 International Conference on Natural Language Processing and Chinese Computing, Qingdao, China, 13–17 October 2021; Springer: Cham, Switzerland, 2021; pp. 548–568. [CrossRef]
45. Zhu, Y.; Lu, S.; Zheng, L.; Guo, J.; Zhang, W.; Wang, J.; Yu, Y. Texygen: A benchmarking platform for text generation models. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 1097–1100. [CrossRef]
46. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318. [CrossRef]
47. Reiter, E. A structured review of the validity of BLEU. *Comput. Linguist.* **2018**, *44*, 393–401. [CrossRef]
48. Wieting, J.; Berg-Kirkpatrick, T.; Gimpel, K.; Neubig, G. Beyond BLEU: Training neural machine translation with semantic similarity. *arXiv* **2019**, arXiv:1909.06694.
49. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. Bertscore: Evaluating text generation with bert. *arXiv* **2019**, arXiv:1904.09675.
50. Hanna, M.; Bojar, O. A fine-grained analysis of BERTScore. In Proceedings of the Sixth Conference on Machine Translation, Online, 10–11 November 2021; pp. 507–517.
51. Meister, C.; Cotterell, R. Language Model Evaluation Beyond Perplexity. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual, 1–6 August 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 5328–5339. [CrossRef]
52. Chen, J.; Zhang, A.; Li, M.; Smola, A.; Yang, D. A cheaper and better diffusion language model with soft-masked noise. *arXiv* **2023**, arXiv:2304.04746.
53. Yuan, H.; Yuan, Z.; Tan, C.; Huang, F.; Huang, S. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *arXiv* **2022**, arXiv:2212.10325.
54. Tang, Z.; Wang, P.; Zhou, K.; Li, J.; Cao, Z.; Zhang, M. Can Diffusion Model Achieve Better Performance in Text Generation? Bridging the Gap between Training and Inference! *arXiv* **2023**, arXiv:2305.04465.
55. Wiseman, S.; Rush, A.M. Sequence-to-sequence learning as beam-search optimization. *arXiv* **2016**, arXiv:1606.02960.
56. Li, C.; Zhang, L.; Zheng, Q.; Zhao, Z.; Chen, Z. User Preference Prediction for online dialogue systems based on pre-trained large model. In Proceedings of the 2023 International Conference on Natural Language Processing and Chinese Computing, Foshan, China, 12–15 October 2023; Springer: Cham, Switzerland, 2023; pp. 349–357. [CrossRef]
57. Jolicoeur-Martineau, A.; Li, K.; Piché-Taillefer, R.; Kachman, T.; Mitliagkas, I. Gotta go fast when generating data with score-based models. *arXiv* **2021**, arXiv:2105.14080.
58. Salimans, T.; Ho, J. Progressive distillation for fast sampling of diffusion models. *arXiv* **2022**, arXiv:2202.00512.
59. Kim, B.; Ye, J.C. Denoising MCMC for accelerating diffusion-based generative models. *arXiv* **2022**, arXiv:2209.14593.