Diffusion Models in NLP: A Survey

Hao Zou, Zae Myung Kim, Dongyeop Kang
University of Minnesota
{zou00080, kim01745, dongyeop}@umn.edu

Abstract

This survey paper provides a comprehensive review of the use of diffusion models in natural language processing (NLP). Diffusion models are a class of mathematical models that aim to capture the diffusion of information or signals across a network or manifold. In NLP, diffusion models have been used in a variety of applications, such as natural language generation, sentiment analysis, topic modeling, and machine translation. This paper discusses the different formulations of diffusion models used in NLP, their strengths and limitations, and their applications. We also perform a thorough comparison between diffusion models and alternative generative models, specifically highlighting the autoregressive (AR) models, while also examining how diverse architectures incorporate the Transformer in conjunction with diffusion models. Compared to AR models, diffusion models have significant advantages for parallel generation, text interpolation, token-level controls such as syntactic structures and semantic contents, and robustness. Exploring further permutations of integrating Transformers into diffusion models would be a valuable pursuit. Also, the development of multimodal diffusion models and large-scale diffusion language models with notable capabilities for few-shot learning would be important directions for the future advance of diffusion models in NLP.

1 Introduction

Diffusion models (Sohl-Dickstein et al., 2015b; Ho et al., 2020; Song et al., 2020) have shown remarkable performance in image generation and attracted huge attention in the field of artificial intelligence. Researchers have also adopted the models to the field of natural language processing (NLP) and have just started to explore their generative capabilities in the domain (Fig. 1). To date, diffusion models have been applied to a wide range of generative NLP tasks, such as unconditional text generation, controllable text generation, machine translation, and text simplification.

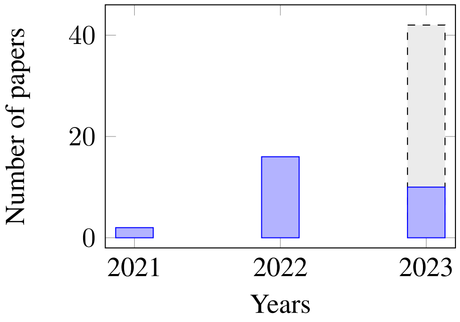


Figure 1: The yearly number of both published and preprinted papers on diffusion models for NLP. For year 2023, the blue bar shows the number collected until the end of April 2023, and the dashed gray bar shows the estimated number for the whole year.

The main challenge in incorporating diffusion models into NLP is the discreteness of texts, which contrasts with the continuous space in which diffusion is modeled. To address this challenge, researchers have introduced modifications to the models, and we categorize them into two approaches:

- **Discrete diffusion models** built on categorical distributions. This method generalizes diffusion process to the discrete domain by corrupting and refining sentences at the token level.
- **Embedding diffusion models** encode discrete texts into continuous space and perform Gaussian noising. As part of this method, additional embedding and rounding steps can be used in the forward and reverse processes, respectively, to convert tokens into embeddings.

In the following sections, we first introduce the general framework of vanilla diffusion models and the modified architecture for discrete state spaces in Section 2. In Section 3, we classify the surveyed architectures into two aforementioned approaches (discrete vs embedding diffusion models), using specific criteria that have been proposed. In Section 4, we conduct a detailed comparative analysis of diffusion models against other generative models in NLP domain. Based on empirical evidence, we highlight the advantages of diffusion models over autoregressive (AR) models, specifically in terms of parallel generation, text interpolation, token-level control, and robustness. In addition, we explore how various surveyed architectures have incorporated the Transformer with diffusion models for NLP. We highlight algorithms and techniques proposed for diffusion models in NLP in Section 5. Finally, we discuss potential future directions that are both timely and worthy of exploration in Section 6.

2 General Framework

Traditionally, diffusion models have focused on continuous state spaces, but recent advancements have expanded their application to discrete state spaces. Discrete diffusion models operate with discrete variables, such as text or categorical data, which present distinct characteristics and challenges.

A key distinction is the treatment of noise. Continuous diffusion models employ additive Gaussian noise, while discrete diffusion models introduce discrete perturbations or transformations to modify the discrete states. This enables exploration of different states and enhances sample diversity.

Transition probabilities also differ between continuous and discrete diffusion models. Continuous models utilize stochastic differential equations, whereas discrete models define transition probabilities using conditional distributions. These distributions capture dependencies between current and previous states, facilitating information propagation and guiding the diffusion process in discrete state spaces.

Diffusion Models Denoising diffusion probabilistic models (DDPMs) were initially introduced by (Sohl-Dickstein et al., 2015a) and enhanced by (Ho et al., 2020). DDPMs employ a two-step process: adding Gaussian noise and performing a reverse process to restore the original data. (Ho et al., 2020) developed DDPMs with an embedding function that maps discrete text to a continuous space, achieving comparable results to state-of-the-art generative models like generative adversarial networks (GANs). Subsequent works (Song et al., 2020; Dhariwal and Nichol, 2021; Nichol and Dhariwal, 2021; Rombach et al., 2021) have further improved the quality and efficiency of DDPMs.

The forward process generates X_{t+1} by adding noise to X_t, creating a dependency solely on X_t. This categorizes the diffusion process as a Markov process, where the noise level is determined by the variance $\beta_t \in (0,1)_{t=1}^{T}$. The expression for $q(x_t|x_{t-1})$ can be written as follows:

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1-\beta_t} \cdot x_{t-1}; \beta_t \mathbf{I}) \quad (1)$$

By applying the reparameterization approach to depict X_t, where $a_t = 1-\beta_t$, $z_t \sim N(0,1)$, $t \leq 0$, the subsequent result can be obtained:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t} Z_{t-1} \quad (2)$$

When computing $q(x_t|x_0)$, the joint probability distribution of $(x_{1:T}|x_0)$ can be determined because it is established as a Markov chain:

$$q(x_{1:T}|x_0) = \sum_{t=1}^{T} q(x_t|x_{t-1}) \quad (3)$$

Then we can express x_t at arbitrary time step t with reference to x_0 in a closed form, where $\bar{\alpha}_t = \alpha_1 \alpha_2 ... \alpha_t$:

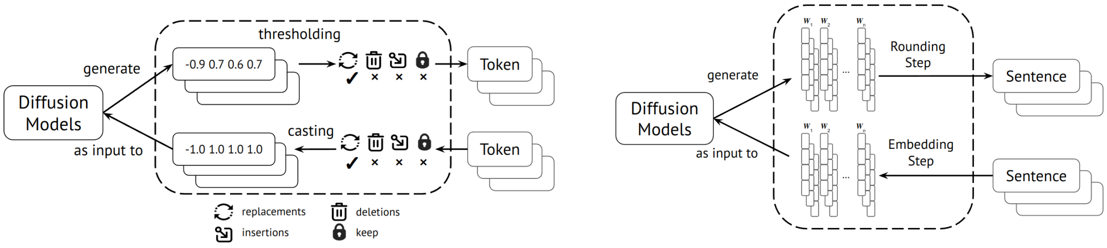
$$q(x_T|x_0) = N(x_t; \sqrt{\bar{\alpha}_t} x_0; (1-\bar{\alpha}_t)\mathbf{I}) \quad (4)$$

For the reverse process, if we can determine the probability distribution of x_{t-1} based on the given condition of x_t, i.e., if $q(x_{t-1}|x_t)$ can be known, then we can iteratively sample random noise to generate an image or sentence. The challenge is to obtain $q(x_{t-1}|x_t)$. To approximate it, we utilize $p_\theta(x_{t-1}|x_t)$. Given that the added noise at each step is relatively small, we assume that $p_\theta(x_{t-1}|x_t)$ follows a Gaussian distribution that can be modeled using a neural network. The reverse process can be expressed as follows:

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu(x_t, t), \sum_\theta (x_t, t)) \quad (5)$$

$$p_\theta(x_{0:T}) = p(x_T) \sum_{t=1}^{T} p_\theta(x_{t-1}|x_t) \quad (6)$$

Applying Bayes' rule, we can express $q(x_{t-1}|x_t, x_0)$ in terms of the known forward conditional probabilities $q(x_t|x_{t-1}, x_0)$, $q(x_{t-1}|x_0)$, and $q(x_t|x_0)$. Our objective is to minimize the mean square error (MSE) loss between the KL divergence of the model p_θ and the true distribution q.



(a) Discrete Diffusion Models

(b) Embedding Diffusion Models

Figure 2: The structures for Discrete Diffusion Models and Embedding Diffusion Models. In Discrete Diffusion Models, the tokens are categorized into categorical values. The figure shows how each token represents a prescribed action to be taken. On the other hand, the Embedding Diffusion Models method involves encoding the entire input sequence into embeddings, followed by applying the diffusion process.

Diffusion models for discrete state spaces For scalar discrete random variables with K categories, where x_t and x_{t-1} take values from 1 to K, the forward transition probabilities can be represented using matrices. Let $[Q_t]_{i,j} = q(x_t = j|x_{t-1} = i)$. We can denote the one-hot representation of x using a row vector, which can be expressed as follows:

$$q(x_t|x_{t-1}) = Cat(x_t; p = x_{t-1}Q_t) \tag{7}$$

In this context, $Cat(x; p)$ represents a categorical distribution over the one-hot row vector x, where the probabilities are determined by the row vector p. The term $x_{t-1}Q_t$ corresponds to a row vector-matrix multiplication. An assumption is made that Q_t is independently applied to each pixel of an image or token in a sequence, and that the distribution q factorizes over these higher dimensions as well. Therefore, we can express $q(x_t|x_{t-1})$ in terms of a single element. Starting from x_0, we can derive the following t-step marginal and posterior at time $t-1$, where $\bar{Q}_t = Q1Q2...Q_t$:

$$q(x_t|x_0) = Cat(x_t; p = x_0\bar{Q}_t) \tag{8}$$

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)} \tag{9}$$

The Markov property of the forward process ensures that $q(x_t|x_{t-1}, x_0)$ can be simplified to $q(x_t|x_{t-1})$. Similarly, assuming the reverse process $p_\theta(x_t|x_{t-1})$ also exhibits a factorized structure, considering the conditional independence of the image or sequence elements, we can derive the KL divergence between q and p_θ by aggregating the probabilities across all possible values of each random variable.

3 A Survey of Diffusion Models in NLP

We present several studies on diffusion models in NLP by grouping them based on their methods for adapting the diffusion process to the textual domain. Specifically, we have two groups: Discrete Diffusion Models and Embedding Diffusion Models (Figure 2). The former operates directly in the discrete input space, while the latter involves lifting discrete inputs into a continuous space.

For each category, we then categorize diffusion models into a multi-perspective taxonomy considering the following criteria: (1) the task they are applied to, (ii) schedule methods during the forward process and (iii) sampling methods used for the reverse process. We note that Reluency in "Schedule" column indicates a linguistic feature that measures the relevance of word w in one sentence d via tf-idf weights. Entropy is a measurement of the amount of information with entropy H in the word w to reflect the importance of that word. Table 1 shows the categorization.

3.1 Discrete Diffusion Models

In the discrete diffusion process, the data is corrupted by switching between discrete values. Discrete diffusion models extend diffusion models to discrete state spaces by corrupting and refining the sentences at the token level.

Multinomial Diffusion (Hoogeboom et al., 2021) introduces a diffusion-based generative model specifically designed for non-ordinal discrete data. It achieves this by diffusing the data to a uniform categorical distribution, effectively capturing the underlying structure while maintaining controlled randomness. The model's transition mechanism

Table 1: Comparison of discrete and embedding diffusion models.

Model	**Tasks**	**Schedule**	**Sampling**
Discrete Diffusion Models			
Multinomial Diffusion (Hoogeboom et al., 2021)	unconditional text generation, unsupervised spell-checking	Transition matrices	—
D3PM (Discrete Denoising Diffusion Probabilistic Models) (Austin et al., 2021b)	char-level text and image generation	Uniform Transition Matrices	—
Zero-shot Diffusion (Nachmani and Dovrat, 2021)	machine translation	Partial Noising	Classifier-free conditional denoising
SUNDAE (Step-unrolled Denoising Autoencoders) (Savinov et al., 2021)	machine translation and unconditional text generation	Uniform Transition Matrices	Low-temperature sampling, Argmax-unrolled decoding, fewer token update
DiffusionBERT (He et al., 2022)	unconditional text generation	Spindle	x0-parameterization
SSD-LM (Semi-autoregressive Simplex-based Diffusion) (Han et al., 2022)	unconditional and controlled text generation	Logits generation	Greedy projection, Sampling, Multi-hot
Bit Diffusion (Generating Discrete Data using Diffusion Models with Self-Conditioning) (Chen et al., 2023b)	categorical image generation and image captioning	—	Self-Conditioning, Asymmetric Time Intervals
DiffusER (Discrete Diffusion via Edit-based Reconstruction) (Reid et al., 2023)	machine translation, summarization, and style transfer	Edit-based Corruption	Beam Search, 2D Beam Search, Nucleus Sampling
Masked-Diffuse LM (Chen et al., 2023a)	controllable text generation	Mask with Entropy and Reluency	Minimum Bayes Risk
RDMs (Reparameterized Discrete Diffusion Model) (Zheng et al., 2023)	machine translation	—	Adaptive Routing Strategy
Embedding Diffusion Models			
Diffusion-LM (Li et al., 2022)	controllable text generation	Cosine	Rounding Step and MBR
DiffuSeq (Gong et al., 2022)	dialogue, question generation, simplification, paraphrasing	Partial Noising	Classifier-free Conditional Denoising, MBR
SED (Self-conditioned Embedding Diffusion) (Strudel et al., 2023)	conditional and unconditional text generation, text infilling	Cosine	Self-conditioning
CDCD (Continuous diffusion for categorical data) (Dieleman et al., 2022)	prompt completion and infilling, machine translation	Partial Noising, Time warping	Self-conditioning, Time warping
Difformer (Gao et al., 2022)	machine translation and abstractive text summarization	Noise Factor	2D parallel decoding
SeqDiffuSeq (Yuan et al., 2022)	dialogue, question generation, simplification, paraphrasing, translation	Adaptive noise schedule	Self-conditioning
DiffuSum (Zhang et al., 2023)	extractive text summarization	—	—
GENIE (Diffusion Language Model Pre-training Framework for Text Generation) (Lin et al., 2023)	text summarization, common sense generation	—	Continuous Paragraph Denoise
DiNoiSer (Diffused Conditional Sequence Learning by Manipulating Noises) (Ye et al., 2023)	machine translation, text simplification, paraphrasing	Manipulated Noises	Self-conditioning, Condition-enhanced Denoiser, Beam Search, Minimum Bayes Risk

involves independent decisions to either resample or retain values, with resampling performed from a uniform categorical distribution.

D3PMs (Austin et al., 2021b) replaces Gaussian noise with Markov transition matrices to diffuse real-world data distribution. It incorporates various types of transition matrices, such as Gaussian kernels, nearest neighbors, and absorbing states, to extend corruption processes. Moreover, D3PMs (Austin et al., 2021b) introduces a novel loss function that combines the variational lower bound with an auxiliary cross-entropy loss. Unlike continuous diffusion, D3PMs (Austin et al., 2021b) allows precise control over the data corruption and denoising process by selecting Q_t in Equation 7, going beyond the use of additive Gaussian noise.

Zero-Shot Diffusion (Nachmani and Dovrat, 2021) utilizes an encoder-decoder architecture with time-based positional encoding for neural machine translation. It employs a transformer encoder to process the source-language sentence and a transformer decoder to handle the noisy target sentence. Notably, this work pioneers conditional text generation using a diffusion model.

Bit Diffusion (Chen et al., 2023b) encodes discrete data as binary bits and trains a continuous diffusion model that treats these binary bits as real numbers. It firstly introduces the self-conditioning technique that greatly improves the sample quality and is widely applied to the following works (Strudel et al., 2023; Dieleman et al., 2022; Yuan et al., 2022).

SUNDAE (Savinov et al., 2021) proposes step-unrolled text generation and is the first non-AR method to show strong results in both machine translation and unconditional text generation.

DiffusER (Reid et al., 2023) employs a 2-dimensional beam search and edit-based text generation. Instead of a pure end-to-end approach, the system divides the task into edit tagging and generation. It generates a sequence of edits to transform a random noise distribution into high-quality output.

DiffusionBERT (He et al., 2022) combines diffusion models with Pre-trained Language Models (PLMs) (Devlin et al., 2018; Lewis et al., 2019; Raffel et al., 2019; Brown et al., 2020; Qiu et al., 2020) by training BERT in reverse of a discrete diffusion process. It introduces a new noise schedule for the forward diffusion process and incorporates the time step into BERT (Devlin et al., 2018). By including the time step, DiffusionBERT captures lost temporal information during diffusion, enhancing the accuracy of the reverse process.

SSD-LM (Han et al., 2022) stands out due to two key features. Firstly, it is semi-autoregressive, enabling iterative generation of text blocks and dynamic length adjustment during decoding. Secondly, it is simplex-based, directly applying diffusion on the natural vocabulary space instead of a learned latent space. This approach facilitates the incorporation of classifier guidance and modular control without the need for modifications to existing classifiers.

Masked-Diffuse LM (Chen et al., 2023a) employs strategic soft-masking, informed by linguistic features, to corrupt both discrete and continuous textual data. It iteratively denoises the data by predicting the categorical distribution. The gradual introduction of perturbations via soft-masking, following an easy-first-generation approach, enhances structural coherence, overall quality, and flexibility in text generation. This pioneering work utilizes linguistic features to effectively corrupt and recover input textual data, improving the generation process.

RDMs (Zheng et al., 2023) introduces a novel reparameterization technique for discrete diffusion models. It employs a stochastic routing mechanism to decide between denoising or noisy resetting for each token. The router ensures uniform processing by assigning equal probabilities to all tokens. This reparameterization simplifies training and enables flexible sampling.

3.2 Embedding Diffusion Models

Recent studies (Li et al., 2022; Gong et al., 2022; Strudel et al., 2023) utilize diffusion processes to generate continuous representations (embeddings) for discrete tokens, known as embedding diffusion models.

Diffusion-LM (Li et al., 2022) constructs diffusion models on continuous word embedding space and incorporates auxiliary losses for joint learning of embedding and network parameters.

DiffuSeq (Gong et al., 2022) focuses on sequence-to-sequence generation using encoder-only Transformers and partial noising to define the diffusion process and learn the denoising function.

SED (Strudel et al., 2023) builds upon the modeling and objectives of Diffusion-LM, introducing a self-conditioning mechanism that enhances baseline performance. Notably, it demonstrates successful scalability to large text datasets like C4 (Raffel et al., 2019).

Difformer (Gao et al., 2022) tackles challenges in applying continuous diffusion models to discrete text generation by addressing denoising objective collapse, imbalanced embedding scales, and inadequate noise during training. It introduces three crucial components: an anchor loss function, layer normalization for embeddings, and an increased noise factor to enhance the scale of added noise.

CDCD (Dieleman et al., 2022) introduces a variance-exploding stochastic differential equations-based diffusion model tailored for text modeling and machine translation. It integrates time warping, an active learning strategy that dynamically adjusts the noise distribution during training to optimize efficiency.

SeqDiffuSeq (Yuan et al., 2022) incorporates self-conditioning and introduces a method to learn token-level noise schedules for text generation. By leveraging appropriate noise schedules, it aims to enhance the quality of generated samples and likelihood modeling (Kingma et al., 2021). In contrast to DiffuSeq (Gong et al., 2022), SeqDiffuSeq (Yuan et al., 2022) explores different model structures and investigates the impact of noise scheduling in sequence-to-sequence tasks.

DiffuSum (Zhang et al., 2023) applies diffusion models to enhance extractive summarization. It generates summary sentence representations and extracts relevant sentences using representation matching. The model introduces a contrastive sentence encoding module that employs matching and multi-class contrastive losses to align and diversify representations. Significantly, DiffuSum represents the first known utilization of diffusion models in the field of extractive summarization.

GENIE (Lin et al., 2023) is a large-scale diffusion-based language model consisting of an encoder and decoder. It enhances noise removal and paragraph-level coherence through continuous paragraph denoise (CPD) loss in pre-training. The CPD objective guides the diffusion-decoder to reconstruct a clean version of a corrupted text paragraph while preserving semantic and syntactic coherence.

DiNoiSer (Ye et al., 2023) addresses small noise effects on "discrete" embeddings in a continuous space, improving diffusion models through noise manipulation in conditional sequence learning. It tackles the discreteness problem by excluding small-scale noises from diffused sequence learner training. For sampling, it introduces an effective method that consistently indicates large noise scales, enhancing the predictive capabilities by amplifying the influence of source conditions on predictions.

3.3 Discrete vs. Embedding Diffusion

	Diffusion Process	**Classifier-based Controls**	**Refinements Adaptation**
Discrete Diffusion	token level	✗	✗
Embedding Diffusion	sequence level	✓	✓

Table 2: Comparative Analysis of Discrete Diffusion Models and Embedding Diffusion Model. Refinements Adaptation column serves as an indicator of the system's ability to incorporate refinements from continuous diffusion in the image domain.

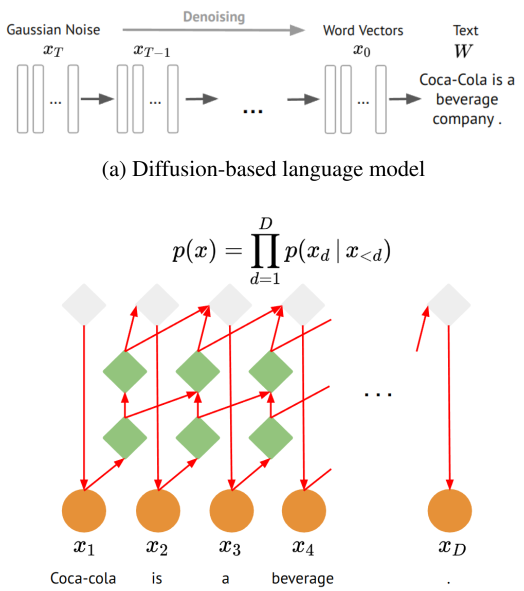
In Table 2, we summarize the advantages of embedding diffusion models over discrete diffusion models.

- **Diffusion Process**: embedding diffusion models transform discrete inputs into a continuous space, enabling representation of multiple outcomes at intermediate timesteps, particularly crucial in capturing token-level uncertainty in language modeling. In contrast, denoising models operating in the discrete input space lack this ability and are confined to specific tokens.
- **Classifier-based Controls**: embedding diffusion models can integrate classifier-based guidance, enhancing the quality of generated samples by leveraging additional information from a classifier to guide the sampling process. In contrast, discrete diffusion models lack this capability, thereby restricting their ability to generate high-quality samples.
- **Refinements Adaptation**: Strudel et al. (2023) showed that discrete diffusion approaches do not reap the advantages derived from the advancements made in continuous diffusion methods within the domain of image processing. Conversely, embedding diffusion models exhibit the capacity to leverage these refinements, rendering them more advantageous and valuable in this context.

4 Diffusion vs. Other Generative Models

4.1 Comparison against Latent Variable Models

Unlike variational autoencoders (VAEs) (Kingma and Welling, 2022; Rezende et al., 2014) or flow-based models (Papamakarios et al., 2017; Kingma and Dhariwal, 2018), diffusion models are learned using a fixed procedure with the latent variable having a high dimensionality (same as the original data). GANs (Goodfellow et al., 2014) are known for potentially unstable training and less diverse generations due to their adversarial training nature.



(a) Diffusion-based language model

(b) AR language model

Figure 3: The comparison between diffusion-based language models and autoregressive language models: diffusion LM iteratively denoises a sequence of Gaussian vectors into word vectors, while AR language model predicts the next word in a sequence of words based on the previous predictions.

VAEs (Kingma and Welling, 2022; Rezende et al., 2014) rely on a surrogate loss. Flow-based models require the construction of specialized architectures to construct reversible transforms.

As Dieleman et al. (2022) notes, diffusion models have a distinct advantage over models like VAEs and GANs, which generate data in a single forward pass. Diffusion models instead focus on reconstructing a small amount of information that has been removed by the corruption process, making the task less challenging.

4.2 Comparison against Autoregressive Models

Autoregressive (AR) models currently dominate the field of language modeling. Also known as causal modeling or the next-token prediction task, AR modeling learns the joint distribution over a token sequence $p(x_1, x_2, ..., x_N)$ by factorizing it into sequential conditionals $p(x_k|x_1, ..., x_{k-1})$ and model them separately with shared parameters (see Figure 3). This means that sampling always proceeds along the left-to-right direction of the sequence. However, in many cases, the ability to go back and refine the earlier parts of the sequence should be useful. In Figure 3, we illustrate the fundamental distinctions between AR and diffusion models, and highlight the distinctive features of the diffusion architecture that endow it with the ability to refine the previous generations, which has potentials to advance the state-of-the-art in the field.

Additionally, Strudel et al. (2023) reveals that, compared to AR models (Bengio et al., 2003; Sutskever et al., 2011; Austin et al., 2021c; Hoffmann et al., 2022), diffusion models can predict all tokens in a sequence at once, which increases interactions between tokens, potentially leading to more coherent samples. Similarly, Savinov et al. (2021) and Li et al. (2022) note that the fixed generation order (left-to-right) from AR models limits the model's flexibility in many controllable generation settings. For example, infilling task, which imposes lexical control on the right contexts, and the syntactic structure control task, which controls global properties involving both left and right contexts. More importantly, this prohibits the iterative refinement of complete text drafts from making them more self-consistent, which is a common task for human writers.

In Table 3, we summarize the empirical benefits of diffusion models over AR models. We categorize them into four aspects: parallel generation, sentence interpolation, token-level control, and robustness to input corruption.

- **Parallel Generation**: diffusion models exhibit a notable departure from the autoregressive nature of AR models. While AR models generate output tokens sequentially conditioned on preceding tokens, diffusion models adopt a parallel generation approach, enabling simultaneous generation of all output tokens. This characteristic enhances the speed and efficiency of text generation, rendering diffusion models particularly suitable for real-time applications.
- **Text Interpolation**: diffusion models demonstrate a superior capacity for text interpolation. Leveraging the denoising process inherent in their design, diffusion models can generate intermediate sentences between two given sentences, ensuring smooth transitions and coherent outputs. This capability enhances the overall fluency and cohesiveness of generated text.
- **Token-level Controls**: Diffusion models provide advanced Token-level Controls, facilitating fine-grained manipulation of generated outputs. This

	Advantages				*Disdvantages*	
Models	**Parallel Generation**	**Text Interpolation**	**Token-level Controls**	**Robustness to Input Corruption**	**Training Complexity**	**Model Interpretability**
AR Models	✗	✗	✗	✗	$\mathcal{O}(n)$	☺
Diffusion Models	✓	✓	✓	✓	$\mathcal{O}(n^T)$	☹

Table 3: Comparative Analysis of Autoregressive (AR) and Diffusion Models in NLP. Token-level Controls of diffusion models include syntactic structure, parse trees, semantic content, parts-of-speech, etc. In terms of training complexity, diffusion models employ multiple rounds of diffusion steps T to generate the entire sequence. Each diffusion step involves optimizing the objective function to capture the denoising process. Specifically, Transformer models are utilized to model the denoising process within each diffusion step.

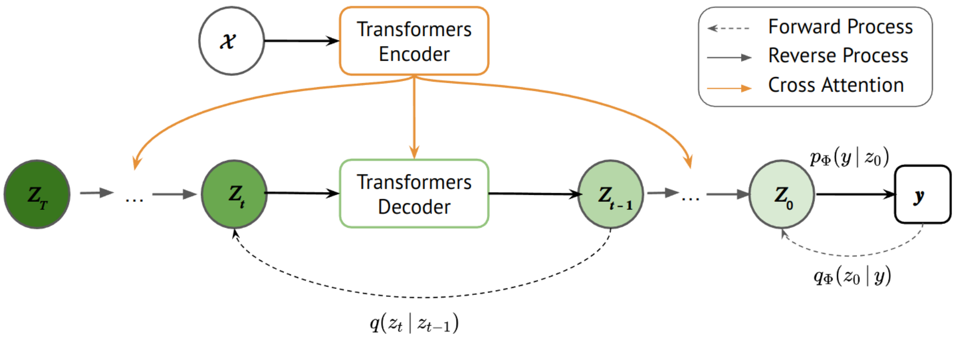


Figure 4: Illustration for how to incorporate Transformers architecture with diffusion models in NLP.

level of control enables precise modifications and interventions in the generated sequences, enhancing the interpretability and applicability of diffusion models in diverse downstream tasks.

- **Robustness to Input Corruption**: Diffusion models exhibit enhanced robustness due to their denoising mechanism that facilitates the reconstruction of the original input. This process aids in mitigating errors and noise present in the input sequence. Consequently, diffusion models are capable of capturing a broader spectrum of input variations by learning a more adaptable distribution over the input data.

In summary, diffusion models offer empirical advantages over AR models, encompassing parallel generation, text interpolation, and advanced token-level controls. These characteristics underscore the potential of diffusion models in various text generation scenarios, emphasizing their efficiency, coherency, and flexibility. In addition to the advantages discussed in Table 3, we also identify two significant disadvantages of diffusion models compared to AR models in terms of training complexity and interpretability.

- **Training Complexity**: Diffusion models are more difficult to train than AR models due to their more complex architecture and optimization objective. In a diffusion model, the entire sequence is generated simultaneously through multiple rounds of diffusion steps, which involve applying a non-linear function to a set of latent variables to obtain the next generation of the sequence. This requires optimizing a complex objective function that includes both the data likelihood and the distance between the generated and ground-truth sequences. On the other hand, AR models generate sequences sequentially by conditioning each time step on the previous ones. This allows for a simpler optimization objective and faster convergence during training.
- **Model Interpretability**: Diffusion models involve multiple non-linear transformations during the diffusion process, resulting in abstract representations in the latent space. These representations may not have a clear interpretation or meaning, and understanding how a specific output sequence is generated from the input can be challenging. This makes diffusion models less interpretable. In contrast, AR models generate sequences step by step, building on the previous steps. Each step is influenced by the preceding steps, making it easier to understand how the output sequence is generated based on the input. AR models are more interpretable

Architecture	Training Corpus	Parameter Size	with Transformer	Pre-trained
Discrete Diffusion Models				
Multinomial Diffusion	text8, enwik8	—	12-layer Transformer	✗
D3PMs	text8, LM1B, CIFAR-10	—	12-layer Transformer	✗
Zero-shot Diffusion	WMT14 (DE-EN, FR-EN) WMT19 (DE-FR)	—	12-layer Transformer	✗
SUNDAE	WMT14 (EN-DE) C4 Python code dataset	63M	encoder-decoder Transformer causality masking removed in the decoder	✗
DiffusionBERT	LM1B	110M	bert-base uncased is trained for reverse process	✓
SSD-LM	OpenWebText	0.4B	bi-directional Transformer encoder a timestep embedding added before the first Transformer block	✗
Bit Diffusion	CIFAR-10, ImageNET MSCOCO 2017	—	6-layer Transformer decoder	✗
DiffusER	WMT'14 CNN/DailyMail, Yelp	—	6-layer Transformer to predict the edit operations 6-layer Transformer for generator	✗
Masked-Diffuse LM	E2E	80M	BERT to encode the input text Transformer to module the reverse process	✓
RDMs (Zheng et al., 2023)	IWSLT14 (DE-EN) WMT14 (EN-DE) WMT16 (EN-RO)	—	Length prediction module on top of Transformer encoder	✓
Embedding Diffusion Models				
Diffusion-LM	E2E, ROCStories	80M	12-layer Transformer	✗
DiffuSeq	CCD, Quasar-T Newsela-Auto Wiki-Auto, QQP	91M	12-layer Transformer	✗
SED	C4	135M & 420M	12-layer Transformer	✓
CDCD	MassiveText, C4 WMT2014, WMT2020	1.3B	Mask-conditional Transformer	✓
Difformer	IWSLT14, WMT14 WMT16, Gigaword	—	6-layer Transformer	✗
SeqDiffuSeq	CCD, Quasar-T, Wiki-Auto QQP, IWSLT14	—	12-layer Transformer	✗
DiffuSum	CNN/DailyMail, XSum PubMed	13M	8-layer Transformer as encoder 12-layer Transformer as generator	✓
GENIE	Gigaword, CNN/DailyMail XSum, CommonGen	—	6-layer Transformer as encoder 6-layer cross attention Transformer as denoising architecture	✓
DiNoiSer	IWSLT14 (DE-EN) WMT14 (EN-DE, EN-RO) Wiki-Auto, QQP	—	12-layer Transformer	✗

Table 4: Training Corpus and connection with Transformer for Discrete and Embedding Diffusion Models. Parameter column refers to the size of used Transformer architecture specifically. Pre-trained column indicates whether the system uses the pre-trained word embedding or not.

due to this sequential nature. These observations highlight the trade-offs associated with diffusion models, emphasizing the need to consider both their advantages and disadvantages in practical applications.

4.3 Transormers with diffusion models

Transformers architecture could be combined with diffusion models, as depicted in Figure 4. Specifically, the Transformer models are used in the encoder-decoder layout to model the denoising function. During the reverse process, the input sequence x therefore only requires one forward computation.

Furthermore, Table 4 provides a comprehensive summary of the training corpus of surveyed systems, highlighting their associations with Transformers. This includes details such as the parameter size and the specific architectures employed by each system for modeling denoising functions, as well as their utilization of pre-trained representations from Transformers during the diffusion process. We hope that this summary can provide researchers with rapid insights into the interplay between Transformers and diffusion models in NLP.

5 Algorithms & Techniques

In this section, we highlight algorithms and techniques proposed for diffusion models in NLP. They are twofold: (1) adapting the models to discrete variables and (2) improving sampling procedures. Figure 5 depicts the algorithms proposed from the surveyed papers.

5.1 Adapting Discrete Variables

5.1.1 Diffusion Steps

To optimize the objective function, DDPM (Ho et al., 2020) utilizes the property that the noise added at each time step in the diffusion process is Gaussian noise; hence the concrete expressions of the objective can be derived. However, the Gaussian distribution here is mainly for continuous domains such as image generations. Hence, D3PM (Austin et al., 2021a) proposed a new method for adding noises for discrete variables. D3PM defined a series of transition matrices that transformed the discrete tokens into `[MASK]` based on pre-defined probabilities at different time steps.

5.1.2 Objective Functions

Predicting initial inputs directly Traditionally, for the approximations of the mean values of each time step, DDPM (Ho et al., 2020) predicts the noise at each time step directly, however, Diffusion-LM (Li et al., 2022) found that the model might fail to generate the initial input x_0 that commits to a single word as the denoising steps cannot ensure that x_0 lies precisely on the embedding of a word. To solve this problem, Diffusion-LM (Li et al., 2022) predicts the initial input x_0 directly in their objective functions.

Partial noising and conditional denoising DiffuSeq (Gong et al., 2022) connects the conditional text c and the target text x, and adds noise only to the target text x in forward process while denoising only x in the denoising process. In contrast to Diffusion-LM's approach (Li et al., 2022) of classifier-guided diffusion, DiffuSeq (Gong et al., 2022) employs a method of classifier-free diffusion that is directed by spatial points. Thus, the system is capable of producing conditional generations in the absence of external classifiers.

5.2 Sampling from Latent Space

Asymmetric Time Intervals Time step plays a critical role in diffusion models. During typical reverse diffusion, symmetric time intervals are often used for both state transition and time reduction, resulting in shared t for $f(x_t, t)$. However, Chen et al. (2023b) shows experimentally that when taking a larger step, using asymmetric time intervals with $f(x_t, t')$, implemented via a simple manipulation of time scheduling at generation, can lead to improved sample quality.

Self-Conditioning When estimating the data sample by the denoising network f at a time step, conditioning the network directly on its previously estimated samples (as opposed to discarding them) can provide better sample quality (Chen et al., 2023b).

Time Warping Dieleman et al. (2022) introduces time warping, an active learning strategy that automatically adapts the distribution of noise levels sampled during training to maximize efficiency. The method alters the relative weighting of the noise levels corresponding to different time steps t. To sample t non-uniformly in practice, the inverse transform sampling can be used: first generate uniform samples $u \in [0, 1]$ and then warp them using the inverse cumulative distribution function (CDF) of the distribution which corresponds to the desired weighting: $t = F - (u)$. This time warping procedure is equivalent to time reweighting in expectation, but more statistically efficient.

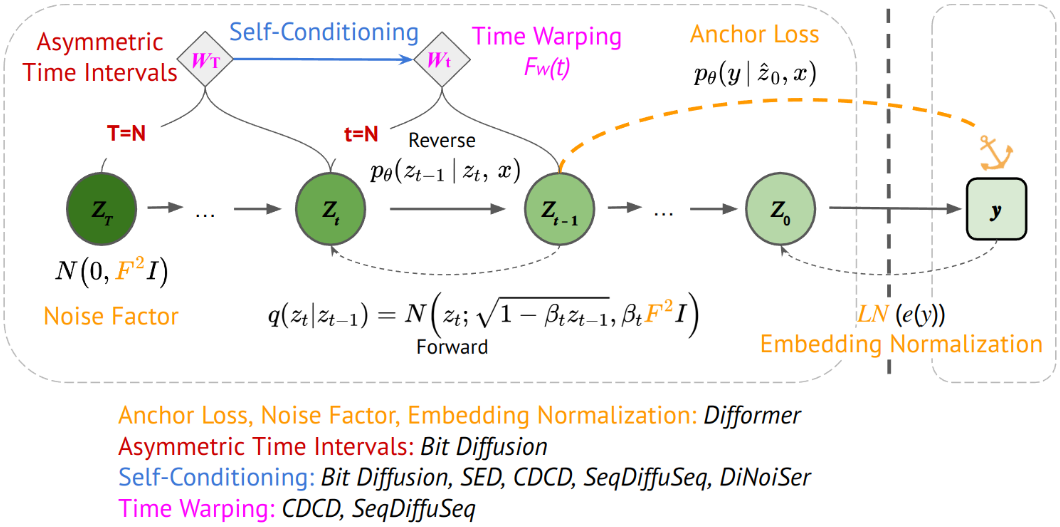

Asymmetric Time Intervals
W_T
Self-Conditioning
W_t
Time Warping
$F_W(t)$
Anchor Loss
$p_\theta(y \mid \hat{z}_0, x)$
T=N
t=N
Reverse
$p_\theta(z_{t-1} \mid z_t, x)$
Z_T
Z_t
Z_{t-1}
Z_0
y
$N(0, F^2 I)$
Noise Factor
$q(z_t|z_{t-1}) = N\left(z_t; \sqrt{1-\beta_t z_{t-1}}, \beta_t F^2 I\right)$
Forward
$LN\ (e(y))$
Embedding Normalization
Anchor Loss, Noise Factor, Embedding Normalization: Difformer
Asymmetric Time Intervals: Bit Diffusion
Self-Conditioning: Bit Diffusion, SED, CDCD, SeqDiffuSeq, DiNoiSer
Time Warping: CDCD, SeqDiffuSeq

Figure 5: Algorithms proposed to adapt the discrete data. Details of the proposed architectures are described in Section 3. Details of the algorithms are described in Section 5.

6 Challenges & Future Directions

In this section, we advance potential lines of inquiry that are both contemporarily significant and intellectually deserving of investigation (Figure 6).

6.1 General Challenges

Latent Space Restriction Diffusion models impose a restriction on the latent space representations, as the dimensions of latent vectors and inputs must be the same. This constraint limits the representational power of the latent vector.

Computational Cost The convergence of diffusion models requires a large number of iterations, which can lead to significant computational costs, especially when dealing with large datasets.

Sensitivity Diffusion models can be very sensitive to the choice of hyperparameters, such as diffusion coefficient, time step size, number of diffusion steps, etc., which can lead to suboptimal performance or even failure to converge.

Dependence on diffusion process assumptions Diffusion models rely on the assumption that information diffuses smoothly and uniformly across the data, which may not always hold in practice. Given perfect mathematical formulation, the diffusion process itself might not be intuitive enough. For instance, optimizing from a totally noisy distribution is quite different to human mind.

Limited interpretability and explainabilities The black-box nature of diffusion models makes it challenging to understand how they make decisions, limiting their interpretability. For instance, the latent vectors learned from diffusion models do not have any linguistic or structural explainabilities.

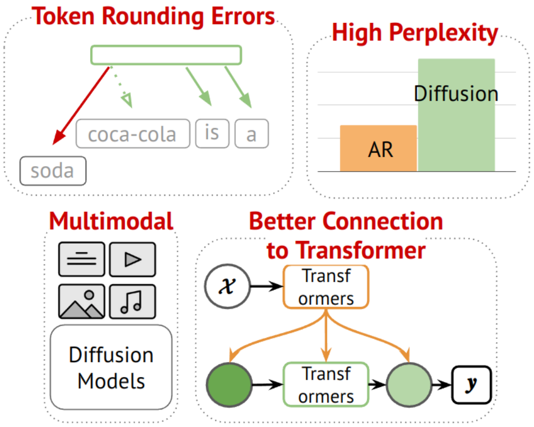


Figure 6: Challenges and future directions we conclude based on the surveyed papers.

6.2 NLP-Specific Challenges

Token Rounding Errors The learned embeddings through embedding diffusion models define a mapping from discrete text to the continuous x_0. We now describe the inverse process of rounding a predicted x_0 back to discrete text. Rounding is achieved by choosing the most probable word for each position. However, empirically, the model fails to generate x_0 that commits to a single word

(Li et al., 2022).

High Perplexity As stated in Li et al. (2022); Lovelace et al. (2022), the perplexity from diffusion models lags behind AR models. However, measuring perplexity with a pretrained AR model such as GPT-2 may bias the metric towards AR models. Besides, previous studies have demonstrated that generating text with low perplexity does not necessarily imply high quality, but rather suggests degenerate behavior. (Nadeem et al., 2020; Zhang et al., 2021). Hence, better metrics which have a stronger correlation with human judgements of quality are needed. For this factor, Pillutla et al. (2021) proposed MAUVE Score, a metric for open-ended text generation that compares the distribution of generated text with that of reference text using divergence frontiers, to better correlate with human judgments.

6.3 Potential Future Directions

More Advanced Ways to connect Transformers How to better combine the spatiality of Transformer and temporality of Diffusion is a tricky question since the ideologies for Transformer and Diffusion are from totally different perspectives. Common architectures from our surveyed paper make The time step t included in the neural net through a Transformer sinusoidal position embedding in each block. And currently people just diffuse the whole sequence of the sentences, diffusion process on single token might be interesting to try on. More variations of injecting Transformers into Diffusion might be needed to explore and deeper analysis is needed with strong foundations.

Large Scaled Diffusion Language Models with impressive few-shot learning capabilities Giant language modeling has made significant strides in recent years and has become a dominant area of research in artificial intelligence. With advances in deep learning and natural language processing, large language models like GPT-3 have shown impressive abilities in tasks such as language translation, text generation, question-answering, and even programming. Currently only SED (Strudel et al., 2023) has studied the scaling issues for diffusion models in NLP, the enormous potential of Large-Scale Diffusion Language Modeling in few-shot learning warrants further exploration.

Multimodal Diffusion Modeling In recent years, there has been a growing interest in developing visual language models (VLMs), which are deep learning models that can understand the relationship between images and natural language. The amazing few-shot performance of VLMs shows great potential to transform how machines interact with the visual world and language, such as Vision-Language Pre-training (ViLBERT) model from Facebook AI Research (FAIR) and the Georgia Institute of Technology, and Flamingo from DeepMind. However, current VLMs are all based on Transformers, the incorporation of Diffusion Models presents vast potential for exploration and discovery.

7 Conclusion

This survey paper extensively discusses the formulations, strengths, limitations, and applications of diffusion models in NLP. We conduct a comprehensive comparison between diffusion models and alternative generative models, focusing on autoregressive (AR) models. Additionally, we explore the integration of the Transformer architecture with diffusion models across various architectures.

Our findings demonstrate the significant advantages of diffusion models over AR models. They excel in parallel generation, enabling faster and more efficient text generation. Diffusion models also demonstrate superior performance in sentence interpolation, token-level controls, and robustness to input corruption. Further research on integrating Transformers into diffusion models and developing multimodal and large-scale diffusion language models for few-shot learning is crucial.

In summary, this survey paper provides a comprehensive overview of diffusion models in NLP, highlighting their benefits, comparative analysis with AR models, and avenues for future research. We hope it can contribute to the understanding and advancement of diffusion models in the field of NLP.

Limitations

The selection of diffusion models included in this paper may introduce a bias based on our knowledge and availability of resources. This could potentially exclude relevant diffusion models that were not considered or well-known at the time of the survey. It is crucial to acknowledge that the selection of specific models and the exclusion of others can impact the comprehensiveness and generalizability of the findings. Another limitation pertains to the understanding and interpretation of the inner workings and decision-making processes of the surveyed diffusion models. Diffusion models in NLP, particularly those employing deep learning techniques, are often regarded as black-box models with limited interpretability. The lack of interpretability can impede the trust and acceptance of diffusion models in practical applications.

Ethics Statement

Diffusion models in NLP may be influenced by biases present in the training data, highlighting the need to consider the ethical implications of deploying biased models in real-world applications. Furthermore, the impact of diffusion models in NLP extends to shaping public opinion, influencing decision-making processes, and affecting social dynamics. Therefore, we prioritize responsible use and communication of the findings in this paper, avoiding sensationalism, misrepresentation, or overgeneralization of the capabilities and limitations of diffusion models in NLP to ensure a well-rounded understanding among the public.

Acknowledgement

This project is supported in part by Sony Research Grant.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021a. Structured denoising diffusion models in discrete state-spaces.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021b. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021c. Structured denoising diffusion models in discrete state-spaces. *ArXiv*, abs/2107.03006.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. In *Journal of machine learning research*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jiaao Chen, Aston Zhang, Mu Li, Alex Smola, and Diyi Yang. 2023a. A cheaper and better diffusion language model with soft-masked noise.

Ting Chen, Ruixiang ZHANG, and Geoffrey Hinton. 2023b. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Prafulla Dhariwal and Alex Nichol. 2021. Diffusion models beat gans on image synthesis.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. 2022. Continuous diffusion for categorical data.

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. 2022. Difformer: Empowering diffusion models on the embedding space for text generation.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. 2022. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control.

Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. 2022. Diffusionbert: Improving generative masked language models with diffusion models.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. 2022. Training compute-optimal large language models. *ArXiv*, abs/2203.15556.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions.

Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. On density estimation with diffusion models. In *Advances in Neural Information Processing Systems*.

Diederik P Kingma and Max Welling. 2022. Auto-encoding variational bayes.

Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. 2023. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Weinberger. 2022. Latent diffusion for language generation.

Eliya Nachmani and Shaked Dovrat. 2021. Zero-shot translation using diffusion models.

Moin Nadeem, Tianxing He, Kyunghyun Cho, and James Glass. 2020. A systematic characterization of sampling algorithms for open-ended language generation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 334–346, Suzhou, China. Association for Computational Linguistics.

Alex Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models.

George Papamakarios, Theo Pavlakou, and Iain Murray. 2017. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Machel Reid, Vincent Josua Hellendoorn, and Graham Neubig. 2023. DiffusER: Diffusion via edit-based reconstruction. In *International Conference on Learning Representations*.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China. PMLR.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models.

Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. 2021. Step-unrolled denoising autoencoders for text generation.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015a. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France. PMLR.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015b. Deep unsupervised learning using nonequilibrium thermodynamics.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models.

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Sussman Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond. 2023. Self-conditioned embedding diffusion for text generation.

Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. Generating text with recurrent neural networks. In *International Conference on Machine Learning*.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. 2023. Dinoiser: Diffused conditional sequence learning by manipulating noises.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. Seqdiffuseq: Text diffusion with encoder-decoder transformers.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Diffusum: Generation enhanced extractive summarization with diffusion.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. 2023. A reparameterized discrete diffusion model for text generation.

Tables

Anonymous ACL submission

Table 1: To classify existing models, we consider three criteria: the task, the denoising condition, and the underlying approach (architecture). Additionally, we list the data sets and evaluation metrics on which the surveyed models are applied. We use the following abbreviations in the architecture column: D3PM (Discrete Denoising Diffusion Probabilistic Models), SUNDAE (Step-unrolled Denoising Autoencoders for Text Generation), DiffusionBERT (Improving Generative Masked Language Models with Diffusion Models), SSD-LM (Semi-autoregressive Simplex-based Diffusion Language Model for Text Generation and Modular Control), Bit Diffusion (Generating Discrete Data using Diffusion Models with Self-Conditioning), DiffusER (Discrete Diffusion via Edit-based Reconstruction), SED (Self-conditioned Embedding Diffusion for Text Generation), CDCD (Continuous diffusion for categorical data)

paper	**Tasks**	**Denoising Condition**	**Architecture**	**Datasets**
Discrete Diffusion Modes				
?	char-level text and image generation	unconditional	D3PMs	text8, LM1B, CIFAR-10
?	unconditional text generation and unsupervised spell-checking	unconditional	Multinomial Diffusion	text8, enwik8
?	machine translation and unconditional text generation	unconditional	SUNDAE	WMT'14, C4, a dataset of Python code
?	unconditional text generation	unconditional	DiffusionBERT	LM1B
?	unconditional text generation and sentiment controlled text generation	unconditional	SSD-LM	OpenWebText
?	categorical image generations and image captioning	unconditional	Bit Diffusion	CIFAR-10, ImageNET; MSCOCO 2017
?	machine translation, summarization, and style transfer	unconditional	DiffusER	WMT'14, CNN/DailyMail, Yelp
Embedding Diffusion Modes				
?	controllable text generation	unconditional, classifier guidance	Diffusion-LM	E2E, ROCStories
?	sequence to sequence text generation	classifier-free conditional	DiffuSeq	CCD, Quasar-T, Newsela-Auto and Wiki-Auto, QQP
?	prompt completion and infilling, machine translation	classifier-free conditional	CDCD	MassiveText, C4, WMT2014, WMT2020
?	machine translation and text summarization	classifier-free conditional	Difformer	IWSLT14, WMT14, WMT16, Gigaword
?	sequenceto-sequence text generation	classifier-free conditional	SeqDiffuSeq	CCD, Quasar-T, Wiki-Auto, QQP, IWSLT14
?	conditional and unconditional text generation	unconditional, classifier-free guidance	SED	C4

Table 2: This table summarizes the techniques each architecture uses or newly proposes for noise schedule and sampling, as well as the evaluation metrics that are applied. We also consider whether the architectures use the pre-trained models or not for further analysis. We use the following abbreviations for the sampling and evaluation metrics columns: PPL (Perplexity), MBR (Minimum Bayes Risk), bpc (bits per character), bpb (bits per raw byte), dist-1 (distinct unigram), div-4 (diverse 4-gram)

paper	**Architecture**	**Pre-trained**	**Schedule**	**Sampling**	**Evaluation Metrics**
Discrete Diffusion Modes					
?	D3PMs	No	$(T-t+1)^{-1}$		NLL, IS, FID, PPL
?	Multinomial Diffusion	No			bpc, bpb
?	SUNDAE	No		Low-temperature sampling, Argmax-unrolled decoding, Updating fewer tokens	BLEU
?	DiffusionBERT	Yes	Spindle	x0-parameterization	PPL, BLEU, Self-BLEU
?	SSD-LM	No	Logits generation	Greedy projection, Sampling, Multi-hot	MAUVE, PPL, Dist-n
?	Bit Diffusion	No		Self-Conditioning and Asymmetric Time Intervals	FID, BLEU-4, CIDEr, ROUGE-L
?	DiffusER	No	Edit-based Corruption	Beam Search, 2D Beam Search, Nucleus Sampling	BLEU, ROUGE
Embedding Diffusion Modes					
?	Diffusion-LM	No	Cosine	Rounding Step and MBR	BLEU, word-level exact match, success rate, human evaluation
?	DiffuSeq	No	Partial Noising	Classifier-free Conditional Denoising, MBR	BLEU, ROUGE, BERTScore, dist-1, self-BLEU, div-4
?	CDCD	No	Time warping	Self-conditioning, Time warping	BLEU, AR-NLL, unigram entropy, MAUVE
?	Difformer	Yes	Noise Factor	2D parallel decoding	BLEU, ROUGE-1/2/L, SacreBLEU
?	SeqDiffuSeq	No	Adaptive noise schedule	Self-conditioning	BLEU, Rouge-L
?	SED	Yes	Cosine	Self-conditioning	AR-NLL, unigram entropy

References