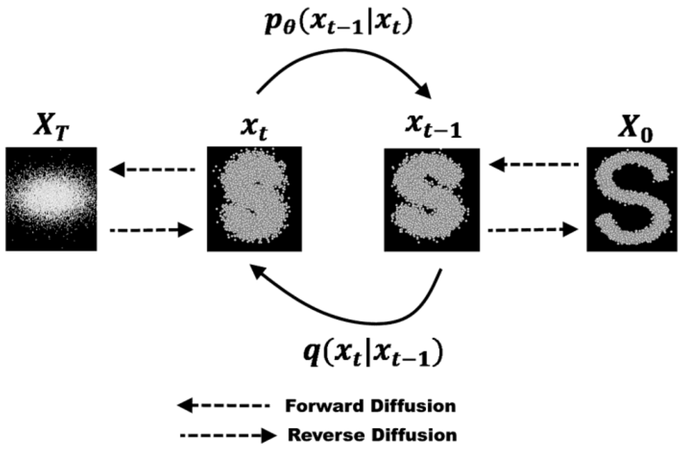
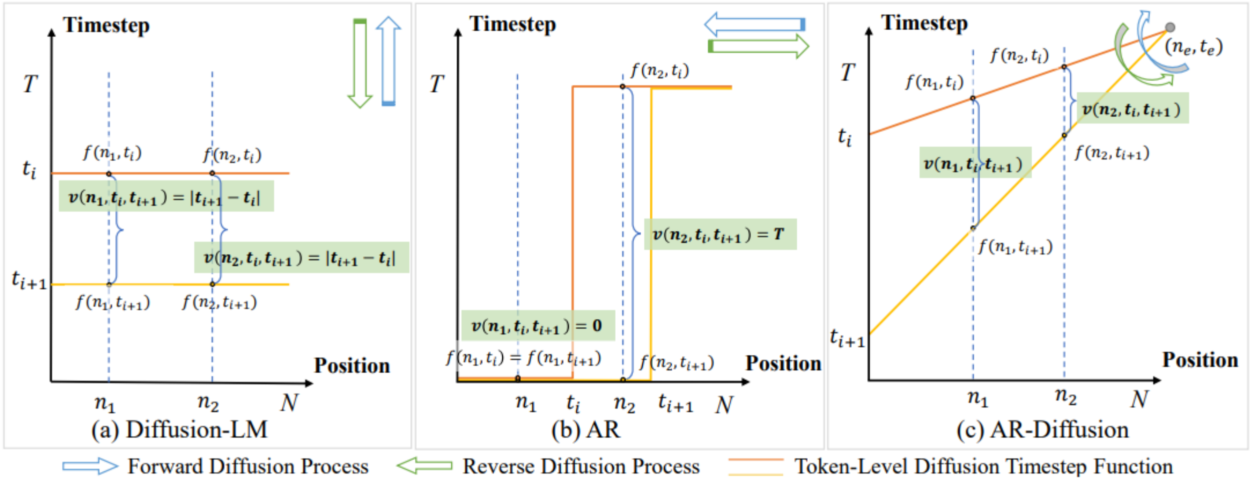
NEURAL INFORMATION PROCESSING SYSTEMS

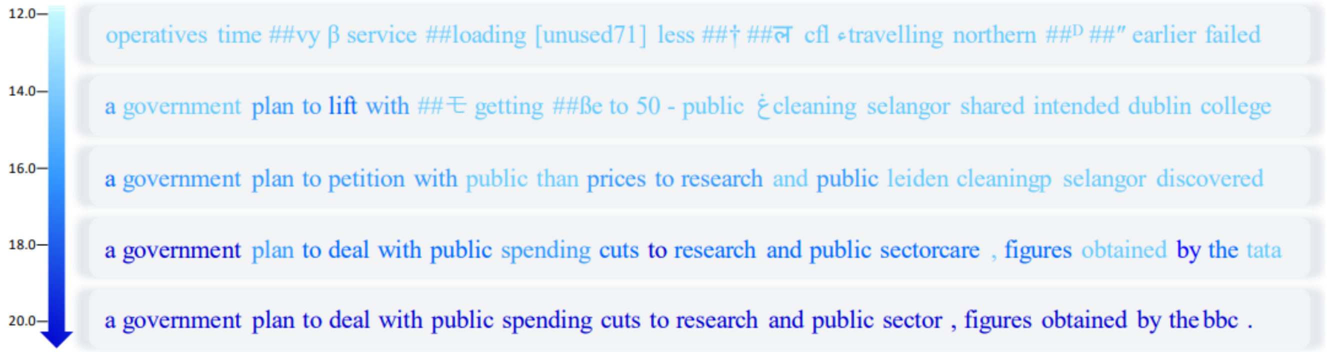
AR-Diffusion: Auto-Regressive Diffusion Model for Text Generation

Tong Wu, Zhihao Fan, Xiao Liu, Yeyun Gong, Yelong Shen, Jian Jiao, Hai-Tao Zheng, Juntao Li, Zhongyu Wei, Jian Guo, Nan Duan, Weizhu Chen

Diffusion Models for Language Generation







How AR-Diffusion works?

Algorithm 1 Training Process of AR-DIFFUSION.

Input: Dataset $\{(\boldsymbol{x}, \boldsymbol{y})\}$, maximum timestep number T and maximum target length N.
Output: Optimized model parameters $\boldsymbol{\theta}$.

Define an anchor point (n_e, t_e)[5].
repeat
  Sample $(\boldsymbol{x}, \boldsymbol{y})$ from the dataset and embed $\boldsymbol{y}$ into $\boldsymbol{z}_0$.
  Sample a sentence-level timestep t from the interval $[0, N+T]$, then the start point is determined by the following equation:

$$(n_s, t_s) = \big(\text{clip}(N - t, 0, N), \text{clip}(t - N, 0, T)\big) \tag{6}$$

  Use the point-slope linear function to determine the token-level timestep $f(n, t)$ in position n:

$$f(n, t) = \text{clip}\Big(\frac{t_e - t_s}{n_e - n_s}(n - n_s) + t_s, 0, T\Big) \tag{7}$$

  Sample $\boldsymbol{z}^n_{f(n,t)}$ for each n in different positions with Gaussian reparameterization.
  According to equation (3) and equation (9), employ gradient descent to optimize the objective:

$$\min_{\theta}\Big[-\log p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{z}_0; \boldsymbol{x}) + \sum_{n=1}^{N}\big\|\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z}^n_{f(n,t)}, f(n,t); \boldsymbol{x}) - \boldsymbol{z}_0\big\|^2\Big] \tag{8}$$

until converged

Algorithm 2 Inference Process of AR-DIFFUSION with the Skipping Mechanism.

Input: Source condition $\boldsymbol{x}$, number of decoding steps M and model parameters $\boldsymbol{\theta}$.
Output: Predicted target embedding $\hat{\boldsymbol{y}}$.

Define an anchor point (n_e, t_e).
Uniformly select a decreasing sequence of timesteps $\{t_i\}_{i=0}^{M}$ ranging from $T+N$ to 0.
Sample $\boldsymbol{z}_{t_0} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
for $i = 0$ to $M-1$ **do**
  Calculate the start point (n_s, t_s) using equation (6).
  Based on the current sentence-level inference steps t_i and the next one t_{i+1}, assign token-level timesteps $f(n, t_i)$ and $f(n, t_{i+1})$ to token in position n using equation (7).
  Reverse sample $\boldsymbol{z}_{t_{i+1}} = \big(\boldsymbol{z}^1_{f(1,t_{i+1})}, \boldsymbol{z}^2_{f(2,t_{i+1})}, \cdots, \boldsymbol{z}^N_{f(N,t_{i+1})}\big)$ from $p_\theta(\boldsymbol{z}_{t_{i+1}} \mid \boldsymbol{z}_{t_i}; \boldsymbol{x})$ with the following formulas:

$$p_\theta(\boldsymbol{z}_{t_{i+1}} \mid \boldsymbol{z}_{t_i}; \boldsymbol{x}) = \prod_{n=1}^{N} p_\theta\big(\boldsymbol{z}^n_{f(n,t_{i+1})} \mid \boldsymbol{z}^n_{f(n,t_i)}; \boldsymbol{x}\big) \tag{10}$$

$$p_\theta\big(\boldsymbol{z}^n_{f(n,t_{i+1})} \mid \boldsymbol{z}^n_{f(n,t_i)}; \boldsymbol{x}\big) \sim \mathcal{N}\big(\boldsymbol{z}^n_{f(n,t_{i+1})}; \lambda \boldsymbol{z}^n_{f(n,t_i)} + \mu \boldsymbol{g}_\theta(\boldsymbol{z}^n_{f(n,t)}, f(n,t); \boldsymbol{x}), \sigma \mathbf{I}\big) \tag{11}$$

end for
Map $\boldsymbol{z}_{t_M}$ to the nearest embedding $\hat{\boldsymbol{y}}$.

Experimental Results

Table 1: Results on XSUM test set. The results of NAR and Semi-NAR are from Qi et al. [2021], and the results of AR are from GLGE [Liu et al., 2021].

Methods	**Pattern**	ROUGE-1	ROUGE-2	ROUGE-L
NAT [Gu et al., 2017]	NAR	24.0	3.9	20.3
iNAT [Lee et al., 2018]	NAR	24.0	4.0	20.4
CMLM [Ghazvininejad et al., 2019]	NAR	23.8	3.6	20.2
LevT [Gu et al., 2019]	NAR	24.8	4.2	20.9
InsT [Stern et al., 2019]	Semi-NAR	17.7	5.2	16.1
iNAT [Lee et al., 2018]	Semi-NAR	27.0	6.9	22.4
CMLM [Ghazvininejad et al., 2019]	Semi-NAR	29.1	7.7	23.0
LevT [Gu et al., 2019]	Semi-NAR	25.3	7.4	21.5
LSTM [Greff et al., 2017]	AR[10]	25.1	6.9	19.9
Transformer [Vaswani et al., 2017]	AR[10]	30.5	10.4	24.2
GENIE [Lin et al., 2023] (k = 50)	Diffusion	29.3	8.3	21.9
AR-DIFFUSION (k = 50)	Diffusion	31.7	10.1	24.7
AR-DIFFUSION (k = 500)	Diffusion	**32.2**	**10.6**	**25.2**

Table 2: Results on CNN/DAILYMAIL test set. The results of AR are from GLGE Liu et al. [2021].

Methods	**Pattern**	ROUGE-1	ROUGE-2	ROUGE-L
LSTM [Greff et al., 2017]	AR	37.3	15.7	34.4
Transformer [Vaswani et al., 2017]	AR	39.5	16.7	36.7
GENIE [Lin et al., 2023] (k = 50)	Diffusion	34.4	12.8	32.1
AR-DIFFUSION (k = 50)	Diffusion	39.6	16.3	37.1
AR-DIFFUSION (k = 500)	Diffusion	**40.2**	**17.1**	**37.7**

Table 3: Results on IWSLT14 DE→EN test set following the setting of SEQDIFFUSEQ. "NFE" indicates the **N**umber of **F**unction **E**valuations [Ye et al., 2023].

Methods	**Pattern**	BLEU	**Steps**	**NFE (Steps$\times k$)**
Transformer [Vaswani et al., 2017]	AR	34.74	-	-
CNAT [Bao et al., 2021]	NAR	29.81	-	-
SeqDiffuSeq [Yuan et al., 2022] (k = 1)	Diffusion	29.83	2,000	2,000 (2,000 $\times$ 1)
AR-DIFFUSION (k = 1)	Diffusion	30.19	20	20 (20 $\times$ 1)
GENIE [Lin et al., 2023] (k = 50)	Diffusion	30.08	20	1,000 (20 $\times$ 50)
AR-DIFFUSION (k = 50)	Diffusion	34.95	20	1,000 (20 $\times$ 50)
AR-DIFFUSION (k = 500)	Diffusion	**35.62**	20	10,000 (20 $\times$ 500)

Table 4: Results on COMMONGEN dev set. Results of NAR and AR are from Lin et al. [2020].

Methods	**Pattern**	ROUGE-2/L		BLEU-3/4		METEOR	SPICE
bRNN-CopyNet [Gu et al., 2016]	AR	9.23	30.57	13.60	7.80	17.40	16.90
Trans-CopyNet [Lin et al., 2020]	AR	11.08	32.57	17.20	10.60	18.80	18.00
MeanPooling-CopyNet [Lin et al., 2020]	AR	11.36	34.63	14.80	8.90	19.20	20.20
LevT [Gu et al., 2019]	NAR	12.22	35.42	23.10	15.00	22.10	21.40
ConstLeven [Susanto et al., 2020]	NAR	13.47	35.19	21.30	12.30	**25.00**	23.20
GENIE [Lin et al., 2023] (k = 50)	Diffusion	12.89	35.21	22.00	13.30	24.30	23.00
AR-DIFFUSION (k = 50)	Diffusion	**13.93**	**37.36**	**25.60**	**16.40**	**25.00**	**24.20**

If you have any questions, please contact

wu-t21@mails.tsinghua.edu.cn