Directed Acyclic Transformer Pre-training for High-quality Non-autoregressive Text Generation

Fei Huang Pei Ke Minlie Huang*
The CoAI group, Tsinghua University, Beijing, China
Institute for Artificial Intelligence, State Key Lab of Intelligent Technology and Systems,
Beijing National Research Center for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing, China
f-huang18@mails.tsinghua.edu.cn, kepei1106@outlook.com,
aihuang@tsinghua.edu.cn

Abstract

Non-AutoRegressive (NAR) text generation models have drawn much attention because of their significantly faster decoding speed and good generation quality in machine translation. However, in a wider range of text generation tasks, existing NAR models lack proper pre-training, making them still far behind the pre-trained autoregressive models. In this paper, we propose Pre-trained Directed Acyclic Transformer (PreDAT) and a novel pre-training task to promote prediction consistency in NAR generation. Experiments on five text generation tasks show that our PreDAT remarkably outperforms existing pre-trained NAR models (+4.2 score on average) and even achieves better results than pre-trained autoregressive baselines in *n*-gram-based metrics, along with 17 times speedup in throughput. Further analysis shows that PreDAT benefits from the unbiased prediction order that alleviates the error accumulation problem in autoregressive generation, which provides new insights into the advantages of NAR generation.[1]

1 Introduction

Pre-trained language models have been widely applied in text generation (Radford et al., 2019; Song et al., 2019; Lewis et al., 2020; Raffel et al., 2020), which can effectively improve the performance of downstream generation tasks, especially in low-resource scenarios (Brown et al., 2020). Most of these pre-trained language models are based on AutoRegressive (AR) generation, which produces high-quality texts by predicting each token one by one. However, such a sequential generation process suffers from high latency and low throughput in inference, thereby largely limiting the use of AR models in scenarios with real-time requirements.

Non-AutoRegressive (NAR) generation is an alternative text generation paradigm (Gu et al., 2018). Unlike sequential generation in AR models, NAR models predict all tokens in parallel, which largely accelerates the decoding process. Although early NAR models suffer from serious quality degradation due to the independent token prediction, recent NAR studies have made much progress on some generation tasks, such as machine translation (Qian et al., 2021; Gu and Kong, 2021; Huang et al., 2022a). Notably, Huang et al. (2022c) propose Directed Acyclic Transformer, which incorporates a directed acyclic graph to reduce the conflicts in capturing possible outputs, achieving a comparable translation quality to the AR models.

Despite the success of NAR generation in machine translation, it is still challenging to apply NAR models to a wider range of generation tasks, mainly due to the lack of appropriate pre-training. Although some previous studies have explored pre-training methods such as directly fine-tuning BERT for NAR generation (Guo et al., 2020b; Su et al., 2021; Jiang et al., 2021) or pre-training NAR models from scratch (Qi et al., 2021; Li et al., 2022), their models still have a significant quality gap compared with AR ones. We argue that these methods do not fully exploit the characteristic of NAR generation, thereby restricting downstream performance. Specifically, we discuss two main issues: (1) Previous pre-training

* Corresponding author: Minlie Huang.

[1] Our code and pre-trained models are available at https://github.com/thu-coai/DA-Transformer.

Transactions of the Association for Computational Linguistics, vol. 11, pp. 941–959, 2023. https://doi.org/10.1162/tacl_a_00582
Action Editor: Alexander Rush. Submission batch: 11/2022; Revision batch: 2/2023; Published 8/2023.

tasks are ineffective in promoting sentence-level prediction consistency, making it hard for their models to predict a whole sentence simultaneously while preserving the fluency in downstream NAR generation. (2) Previous pre-training tasks fail to address the multi-modality problem (Gu et al., 2018), which has proved to be a fundamental and important challenge in training NAR models (Huang et al., 2022b).

In this paper, we introduce PreDAT, a Pre-trained Directed Acyclic Transformer for high-quality non-autoregressive text generation. We utilize the architecture of Directed Acyclic Transformer and further propose a novel pre-training task, Double-Source Text Infilling (DSTI), aiming to address the above issues in pre-trained NAR models. Specifically, DSTI contains two steps: It corrupts a sentence and scatters the tokens into two sequences, which are fed into the encoder and decoder as two sources of information; then the model is trained to recover the corrupted fragments non-autoregressively. During the pre-training, our model predicts long sentence fragments (about 15 tokens) from nearby contexts, which promotes prediction consistency and bidirectional dependencies. Moreover, DSTI designs a strategy for creating pre-training data pairs that allow the output sequences to have flexible lengths, which well incorporates various alignment-based NAR training objectives to alleviate the multi-modality problem (Libovický and Helcl, 2018; Ghazvininejad et al., 2020; Du et al., 2021; Huang et al., 2022c).

Automatic evaluation shows that PreDAT is effective and efficient on five text generation tasks. It remarkably outperforms previous pre-trained NAR models (+4.2 score on average) and even achieves better results than pre-trained AR baselines in n-gram-based metrics (+0.7 score on average), along with a 17x speedup in throughput. To our knowledge, PreDAT is the first NAR model that outperforms pre-trained AR models on various generation tasks in automatic evaluation. Further ablation studies verify that our pre-training task designs, including the long fragment prediction and alignment-based training objectives, are crucial for success.

To better understand the advantages and weaknesses of NAR generation, we use automatic and manual methods to investigate the generated texts in downstream tasks. We find that PreDAT can alleviate the error accumulation in AR generation and improve the relevance to the input, thereby leading to a better performance in n-gram-based metrics. However, we also find that NAR models, including PreDAT, are still weaker than AR models in preserving the consistency among generated tokens, leading to grammatical errors such as wrong word choices. We believe that these findings can provide novel insights for future NAR studies.

2 Related Work

Pre-trained Language Models (PLM) In recent years, PLMs have made significant progress in natural language generation (Radford et al., 2019; Song et al., 2019; Lewis et al., 2020; Raffel et al., 2020). These PLMs are pre-trained on a large corpus of unlabeled data, where the knowledge can be transferred to downstream tasks, resulting in improved generation quality.

Non-Autoregressive Generation Although NAR generation (Gu et al., 2018) remarkably speeds up the inference, Huang et al. (2022b) point out that it theoretically suffers from serious information dropping, previously known as the multi-modality problem. To alleviate the problem, previous studies propose methods including (1) iterative refinement (Lee et al., 2018; Gu et al., 2019; Ghazvininejad et al., 2019; Guo et al., 2020a; Huang et al., 2022d); (2) knowledge distillation (Kim and Rush, 2016; Ding et al., 2022, 2021a,b; Shao et al., 2022); (3) dependency enhancement (Sun et al., 2019; Qian et al., 2021; Huang et al., 2022a; Bao et al., 2022); or (4) alignment-based objectives (Ghazvininejad et al., 2020; Du et al., 2021; Libovický and Helcl, 2018; Huang et al., 2022c).

There are also studies combining PLMs and NAR generation. For example, some methods fine-tune existing pre-trained models directly (Jiang et al., 2021) or with an adapter (Guo et al., 2020b; Su et al., 2021). Some others combine AR and NAR prediction (Qi et al., 2021) or involve an early exiting mechanism (Li et al., 2022) in pre-training.

Compared with these studies, our method has two significant differences: (1) Previous methods either predict short spans (e.g., BERT) or incorporate unidirectional AR prediction (Qi et al., 2021), which hardly contribute to NAR generation that predicts a whole sentence with bidirectional attention. In contrast, we train our model

to predict long fragments simultaneously, leading to better consistency among generated tokens. (2) Previous methods use a token-level loss that forces the model to predict a same-length sequence to match the target, which over-penalizes the position shift error (Ghazvininejad et al., 2020) and worsens the multi-modality problem. We introduce an up-sampling strategy to obtain longer output sequences, which well incorporates previous alignment-based NAR losses to address the above problems.

3 Preliminaries: Directed Acyclic Transformer

Directed Acyclic Transformer (DAT, Huang et al., 2022c; see also Figure 1) is an NAR model that effectively alleviates the multi-modality problem. It introduces a longer decoding sequence and an alignment-based objective to reduce the conflicts in capturing multiple possible outputs. Specifically, given the input $X = \{x_1, \cdots, x_M\}$ and the target sequence $Y = \{y_1, \cdots, y_N\}$, DAT produces a feature sequence $V = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_L\}$ organized in a Directed Acyclic Graph (DAG), where Y is aligned to a sub-sequence of V (equivalently, assigned to a path of the DAG). Notably, L is usually much larger than N to allow for more flexible alignments. In DAT training, the alignment-based objective marginalizes the probabilities of all possible alignments that produce the target Y, formally as

$$\begin{aligned}\mathcal{L}_{\mathrm{DAT}}(V, Y) &= -\log P_\theta(Y|X) \\ &= -\log \sum_{A \in \Gamma} P_\theta(Y|A, X) P_\theta(A|X), \quad (1) \\ P_\theta(Y|A, X) &= \prod_{i=1}^{L} P_\theta(y_i|\mathbf{v}_{a_i}), \\ P_\theta(A|X) &= \prod_{i=1}^{L-1} P_\theta(a_{i+1}|a_i),\end{aligned}$$

where $A = \{a_1, \cdots, a_L\}$ is feature indexes on the aligned path, and Γ contains all possible paths with the size of $\binom{L}{N}$. $P_\theta(y_i|\mathbf{v}_{a_i})$ represents token probabilities predicted from the feature $\mathbf{v}_{a_i}$, and $P_\theta(a_{i+1}|a_i)$ represents transition probabilities revealing how likely a_{i+1} follows a_i in a path. Since it is impossible to enumerate the huge number of paths in Equation 1, a dynamic programming algorithm can be adopted to address the problem, whose details can be found in the original paper (Huang et al., 2022c).

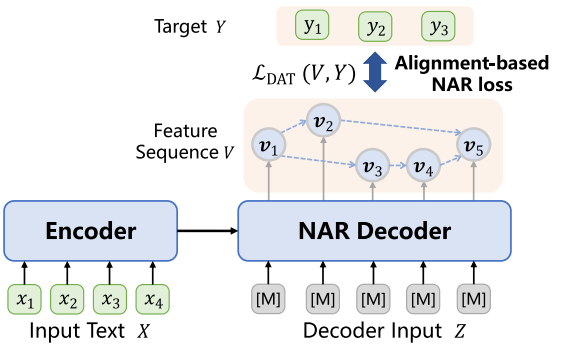

Figure 1: Preliminaries: Directed Acyclic Transformer (DAT). To alleviate the multi-modality problem, DAT predicts a feature sequence V organized in a directed acyclic graph (DAG) and then adopts an alignment-based objective that aligns the target Y to the feature sequence V, represented by $\mathcal{L}_{\mathrm{DAT}}(V, Y)$.

Compared with previous NAR models, DAT explicitly models the dependencies between tokens by the position transitions and is able to store multiple modalities on different paths of the DAG, thereby remarkably improving the generation performance. Moreover, various decoding algorithms such as beam search and Nucleus sampling (Holtzman et al., 2020) can be utilized to boost the generation quality or diversity.

Besides $\mathcal{L}_{\mathrm{DAT}}$, there are other alignment-based objectives that succeed in alleviating the multi-modality problem in NAR generation, such as AXE (Ghazvininejad et al., 2020), OaXE (Du et al., 2021), and CTC (Graves et al., 2006; Libovický and Helcl, 2018). In general, these objectives are also obtained by aligning the target Y with the feature sequence V, thus denoted by $\mathcal{L}(V, Y)$.

4 Proposed Method

In this section, we introduce PreDAT, Pretrained Directed Acyclic Transformer. We first propose the pre-training task (Section 4.1) and then describe the fine-tuning and inference strategies (Section 4.2).

4.1 Pre-training Task

Our pre-training task, Double-Source Text Infilling (DSTI), is a self-supervised pre-training task that aims to promote prediction consistency and bidirectional dependencies for NAR models. Our task scatters part of a sentence into two

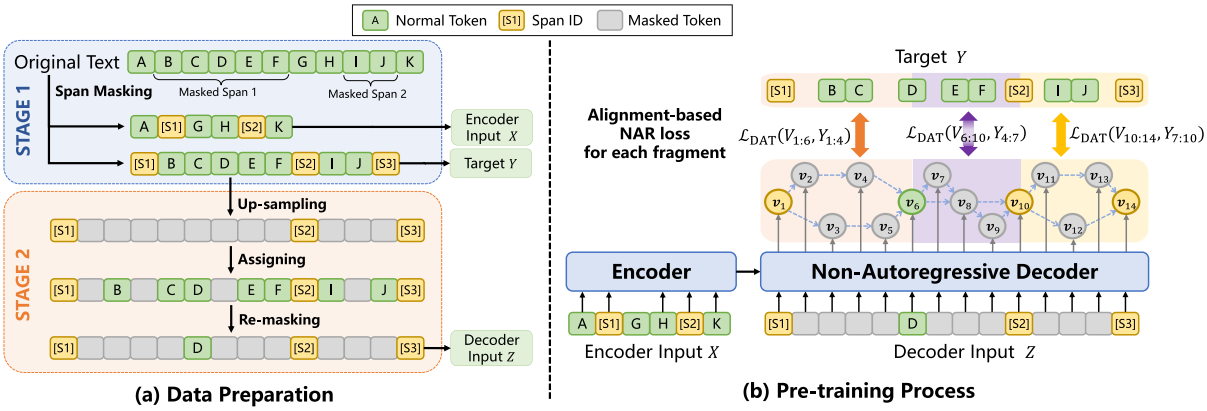

Figure 2: An overview of Double-Source Text Infilling (DSTI). (a) Data Preparation: DSTI first creates the encoder input X and the target Y by span masking, and then obtains the decoder input Z by up-sampling, assigning, and re-masking. (b) Pre-training Process: The NAR model is trained to predict the unseen fragments in Y in parallel, with X and Z as inputs. The training objective is the sum of alignment-based NAR losses, which are obtained by aligning each target fragment (e.g., $Y_{1:4}$) to the feature sequence on the corresponding masked segments (e.g., $V_{1:6}$).

sequences, feeds them into the encoder and decoder as two sources of information, and then trains the model to predict long unseen fragments in a non-autoregressive fashion. Although DSTI is compatible with various NAR architectures and losses, we mainly focus on DAT due to its superior performance.

As shown in Figure 2, our task takes a piece of text from the pre-training corpus and decomposes it into a triple (X, Z, Y), where $X = \{x_1, \cdots, x_M\}$ is the encoder input, $Z = \{z_1, \cdots, z_L\}$ is the decoder input, and $Y = \{y_1, \cdots, y_N\}$ is the target. The data preparation consists of two stages.

Stage 1: Creating Encoder Input We utilize span masking (Raffel et al., 2020) to obtain the encoder input X and the target Y. Specifically, we randomly mask tokens in the original sentence, and then replace consecutive masks into a single special token representing the span ID. Then the prediction target Y is constructed by concatenating the masked spans with the span IDs as delimiters.

Specially, we force each masked span to be long enough (about 15 tokens) because the NAR model has to generate a whole sentence simultaneously in inference, where predicting short spans is unhelpful in preserving sentence-level consistency.

Stage 2: Creating Decoder Input The decoder input Z plays two roles in our pre-training: (1) It reveals some target tokens to promote bidirectional dependencies in the decoder. (2) It determines the length of the predicted feature sequence.

To incorporate the alignment-based NAR losses that require a longer feature sequence than the target (such as DAT and CTC), we create the decoder input Z by an up-sampling step. Then we assign a part of the target tokens to appropriate positions in Z, where the unseen tokens will be used as prediction targets. Specifically, creating Z follows three steps: up-sampling, assigning, and re-masking.

For **up-sampling**, we decide the length of Z based on the target length. Formally, we have $L := \lambda N$, where λ is an up-sampling ratio. In DAT, varying L can bring different DAG sizes and structures, where we sample λ from a uniform distribution to diversify the DAG structures in pre-training. After determining the length, the span IDs are put into Z according to the up-sampling ratio, which will not be modified in the later steps.

For **assigning**, we distribute the target tokens in Z, regardless of whether the token will appear in the final input. Formally, we use an assignment sequence $\{a_i\}_{1 \le i \le N}$ indicating that $z_{a_i} := y_i$. All other positions in Z are masked. For obtaining the sequence $\{a_i\}$, a straightforward strategy is to use uniform assignment, such that every two consecutive target tokens are separated by a constant number of [Mask]. In the pilot experiment, we find it better to use the strategy of glancing training (Huang et al., 2022c; Qian et al., 2021),

which first predicts a DAG with a fully masked Z and then assigns the target tokens on the positions that form the most probable path of the DAG.

For **re-masking**, we determine the tokens finally appearing in Z and then mask the remaining ones. Formally, we randomly sample a fixed proportion of tokens to form a set R, where $z_{a_i} := y_i$ if $i \in R$, and all the other tokens in Z are masked.

Training Objective Our objective is to reconstruct the unseen target fragments according to the given context, similar to masked language modelling (Devlin et al., 2019) but with a significant difference. Instead of using a token-level loss that forces each masked position to predict the corresponding target token, we obtain the sum of alignment-based losses that aligns each unseen target fragment to the feature sequence predicted on the corresponding masked segments. Note that the feature sequence is longer than the target fragment, which brings a larger DAG with a higher capacity to capture multiple possible infilling results.

Specifically, the decoder input consists of several consecutive masked segments segmented by the observed token or span IDs. Each masked segment will produce a feature sequence $V_{a_i:a_j}$, which is then aligned to the corresponding target fragments $Y_{i:j}$ for the DAT loss. The final loss is equal to the sum of the DAT loss of all fragments. Formally,

$$V = \left[\mathbf{v}_1, \cdots, \mathbf{v}_{|Z|}\right] = f_\theta(X, Z),$$
$$\mathcal{L} = \sum_{i,j \in \text{frag}(R)} \mathcal{L}_{\text{DAT}}(V_{a_i:a_j}, Y_{i:j}),$$

where $\text{frag}(R)$ consists of all pairs (i, j) representing the start and end position of unseen fragments, and $\mathcal{L}_{\text{DAT}}$ is defined in Equation (1).

Notably, our idea can be applied to other alignment-based NAR losses, such as CTC loss (Graves et al., 2006), which also trains the model by aligning the target fragment to a longer predicted feature sequence. We verify the generality of DSTI with various loss functions in Section 5.4.

4.2 Fine-tuning and Inference

We generally follow the original training method (Huang et al., 2022c) to fine-tune our PreDAT on the downstream datasets while introducing some improvements: We add a target length predictor for better adaption to tasks with various ratios of input and target lengths, and further propose a trick to improve the decoding throughput.

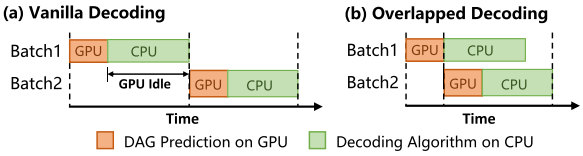

Figure 3: Illustrations of (a) vanilla decoding and (b) overlapped decoding. Overlapped decoding reduces the GPU idle time, leading to higher decoding throughput.

Length Prediction The original DAT simply sets the feature length L to be a constant multiple of the input length, which in most cases of machine translation, satisfies the constraint that the feature sequence should be longer than the target length. However, the targets in our downstream tasks can be arbitrarily long, making this strategy improper.

To better apply PreDAT to various generation tasks without the constraint of input and target length, we introduce a length predictor during fine-tuning and inference. Specifically, in fine-tuning, we use a similar up-sampling strategy as the pre-training to obtain the decoder input length, i.e., λ times the target length. Then we adopt a length predictor on the top of the encoder and train it to predict the target length as a classification. In inference, we obtain the predicted length from the predictor, and then multiply it with $\hat{\lambda}$ to obtain the decoder input length, where $\hat{\lambda}$ is a hyperparameter tuned on the validation set that controls the length of generated sentences.

Overlapped Decoding PreDAT predicts the DAG in parallel on GPU, and then executes a decoding algorithm (e.g., beam search; Huang et al., 2022c) on CPUs to obtain the most likely output from the DAG. As shown in Figure 3, we overlap the GPU and CPU execution, which reduces the GPU idle time and utilizes multiple CPU cores to parallelly process the batches, leading to remarkably higher decoding throughput while not affecting the latency.

5 Experiments

5.1 Implementation Details

Model Configurations Our PreDAT is based on a 6-layer encoder-decoder Transformer (Vaswani

et al., 2017) with a hidden size of 768, following the base version of AR and NAR baselines.

Pre-Training We pretrain PreDAT with DSTI on 16GB English corpus from Wikipedia and BookCorpus (Zhu et al., 2015), with the vocabulary of *bert-base-uncased*. In stage 1, we take a sequence with about 600 tokens and mask 6 equal-length spans that account for 15% tokens. In stage 2, we sample λ uniformly from $[4, 8]$ and mask 90% tokens in the re-masking step. Unless otherwise specified, we pre-train PreDAT for 500k update steps with a batch size of 256 samples and use Adam optimizer (Kingma and Ba, 2015) with a learning rate of 2e-4. We utilize LightSeq (Wang et al., 2022) to accelerate the training (not used in inference), and the pre-training lasts approximately 72 hours on 8 Nvidia A100-40G GPUs.

Fine-Tuning We fine-tune PreDAT on downstream datasets with the DAT loss and glancing training (Qian et al., 2021; Huang et al., 2022c) without knowledge distillation. According to the average sample lengths of each dataset, each mini-batch has approximately 4k target tokens for PersonaChat, XSUM, SQuAD1.1, and 8k target tokens for ROCStory and Quora. We use the early-stop trick according to the performance on the validation set. It usually takes less than 60k steps on SQuAD1.1, Quora, and PersonaChat, and 100k steps on XSUM and ROCStory. We tune the glancing ratio from {0.3, 0.5}, and learning rate from {1e-5, 2e-5, 5e-5, 1e-4, 2e-4}. We evaluate the model every 5k steps on the validation set and obtain the final model by averaging the five best checkpoints.

Inference We utilize lookahead decoding (default unless otherwise specified) and beamsearch (Huang et al., 2022c) to decode a sequence from predicted DAG. We use a beam size of 200 and incorporate a 5-gram LM in the beam search. For open-ended generation, we further employ Nucleus sampling (Holtzman et al., 2020).

For these three decoding strategies, we prevent any repeated tri-gram in expanding the decoding path on the DAG, which is inspired by a similar strategy used in autoregressive decoding (Paulus et al., 2018). Moreover, we also prevent consecutive uni-gram and bi-gram repetitions, which are common errors in PreDAT's outputs.

Dataset	Task	# Samples	Length
SQuAD1.1♠	Question Generation	75k/10k/12k	149.4/11.5
XSUM♠	Summarization	204k/11k/11k	358.5/21.2
Quora♡	Paraphrase Generation	138k/5k/4k	11.5/11.5
PersonaChat♠	Dialog Generation	122k/15k/14k	120.8/11.8
ROCStory♣	Story Generation	88k/5k/5k	9.2/41.6

Table 1: Dataset statistics. **# Samples** shows the number of samples in training/validation/test set. **Length** shows the average length of input/target. We use the processed datasets and evaluation metrics from ♠ Liu et al. (2021), ♡ Jiang et al. (2021), ♣ Guan et al. (2020).

5.2 Experiment Settings

Datasets and Metrics We utilize five datasets: SQuAD1.1 (Rajpurkar et al., 2016), XSUM (Narayan et al., 2018), Quora[2], PersonaChat (Zhang et al., 2018), and ROCStory (Mostafazadeh et al., 2016). We use the processed datasets and the evaluation metrics from previous work, as shown in Table 1. Note that we use corpus BLEU (Papineni et al., 2002) on all datasets because the sentence BLEU may unreasonably prefer very long outputs due to the smoothing method.[3]

To evaluate the decoding speedup, we use two metrics: Latency measures the average time of processing a single sample, and throughput measures the average speed in processing the whole test set, where we tune the batch size to maximize the throughput. All models except MIST are implemented with Fairseq (Ott et al., 2019) + Apex, where MIST is implemented with HuggingFace's Transformers (Wolf et al., 2019). For the beam search algorithm on DAG, we adopt the C++ implementation provided by Huang et al. (2022c). The C++ optimization only affects the extra decoding step on the CPU, but does not speedup the transformer model. All results of speed are evaluated on a workstation with an Nvidia V100-32G GPU and 2 Intel Xeon Gold 6226R CPUs with 32 cores.

Baselines Our baselines include autoregressive Transformer (Vaswani et al., 2017), pre-trained AR models (MASS, Song et al., 2019; BART, Lewis et al., 2020; ProphetNet, Qi et al., 2020), non-pretrained NAR models (Vanilla NAT,

[2]https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs.

[3]Some previous work (Liu et al., 2021) utilize nltk's sentence BLEU with SmoothingFunction().method7.

Model	Pre-trained?	SQuAD1.1			XSUM			Quora			Avg.	Latency		Throughput	
		R-L	B-4	MTR	R-1	R-2	R-L	B-1	B-4	MTR		ms/sample		samples/s	
Autoregressive Text Generation Models															
Transformer	N	29.43	4.61	9.86	30.66	10.80	24.48	58.57	30.14	31.79	25.59	–		–	
MASS	Y	49.48	20.16	24.41	39.70	17.24	31.91	60.56	32.39	32.92	34.31	353	(1.0x)	12	(1.0x)
BART	Y	42.55	17.06	23.19	38.79	16.16	30.61	61.56	31.57	32.42	32.66	–		–	
ProphetNet	Y	48.00	19.58	23.94	39.89	17.12	32.07	62.59	33.80	33.95	34.55	–		–	
Non-autoregressive Text Generation Models															
Vanilla NAT	N	31.51	2.46	8.86	24.04	3.88	20.32	39.85	9.33	18.90	17.68	–		–	
GLAT+CTC	N	30.31	3.21	10.21	31.34	9.06	24.68	58.96	26.67	30.55	25.00	24	(14.7x)	267	(21.5x)
DSLP+CTC	N	28.70	3.00	10.59	28.75	7.35	22.73	61.12	29.70	32.37	24.92	24	(14.7x)	265	(21.4x)
LatentGLAT	N	28.28	2.38	10.43	28.44	7.00	22.66	59.78	28.30	31.26	24.28	28	(12.8x)	334	(27.0x)
BANG	Y	44.07	12.75	18.99	32.59	8.98	27.41	55.18	24.97	25.99	27.88	**18**	**(19.6x)**	**360**	**(29.0x)**
MIST	Y	47.13	16.00	21.10	34.63	11.29	28.70	59.65	29.00	31.56	31.01	22	(15.9x)	159	(12.8x)
PreDAT (Ours)	Y	49.78	21.74	24.58	38.80	16.07	31.78	**62.63**	32.59	33.37	34.59	26	(13.8x)	278	(22.5x)
w/ BeamSearch	Y	**50.41**	**22.66**	**25.11**	**39.79**	**17.38**	**32.71**	62.62	**33.18**	**33.52**	**35.26**	63	(5.7x)	214	(17.3x)
w/o Pre-training	N	30.11	3.30	10.32	32.56	11.17	26.21	59.82	28.17	31.10	25.86	25	(14.3x)	272	(21.9x)

Table 2: Performance on closed-ended text generation datasets. **Bold** and underlined values indicate the best methods in NAR models and all models, respectively. **Latency** measures the average time of processing samples with a batch size of 1, and **Throughput** measures the speed of processing samples with a large batch size (tuned to maximize the throughput), which are evaluated on the test set of XSUM. The metrics include ROUGE-1/2/L (R-1/2/L), BLEU-1/4 (B-1/4), and METEOR (MTR).

Gu et al., 2018); GLAT+CTC, Qian et al., 2021; DSLP+CTC, Huang et al., 2022a; LatentGLAT, Bao et al., 2022), and pre-trained NAR models (BANG, Qi et al., 2021; MIST, Jiang et al., 2021). All these baselines have the same number of layers and hidden sizes as our PreDAT, except that LatentGLAT utilizes a 4-layer latent predictor and a 4-layer decoder based on the original implementation. Note that CTC-based models also require an up-sampling strategy, so we add a length predictor following the description of Section 4.2. Their up-sampling ratio is sampled from $[1.5, 2]$ in training and tuned on the validation set in inference. For AR baselines, unless otherwise specified, we use BeamSearch with a beam size of 5 and the tri-gram repetition prevention trick (Paulus et al., 2018), and tune the length penalty on the validation set. For NAR baselines, we use greedy decoding and further remove consecutive repeated tokens after generation (Li et al., 2019). Some results are collected from Liu et al. (2021); Qi et al. (2021); Jiang et al. (2021).

5.3 Automatic Evaluation

Closed-Ended Text Generation We first test PreDAT on three closed-ended text generation tasks, including question generation, summarization, and paraphrase generation. Closed-ended text generation tasks usually have strict semantic constraints on the outputs, aiming to test the model's ability to extract and organize information.

As shown in Table 2, PreDAT achieves surprisingly good results in both speed and quality. We highlight our advantages as follows:

- PreDAT remarkably improves the quality of NAR generation. Compared with previous pretrained NAR models, PreDAT brings large improvement (+4.2 scores on average) due to our DSTI pre-training and the DAT architecture. Moreover, PreDAT even outperforms the best AR baseline by 0.7 scores. To our knowledge, it is the first time that an NAR model achieves comparable and even stronger performance than AR models in *n*-gram-based metrics on a wide range of text generation tasks.
- PreDAT is highly efficient. Although our model is slightly slower than previous NAR models due to a longer sequence prediction, it still achieves a speedup of 5∼14 times in latency and 17∼23 times in throughput compared with AR generation. It verifies that PreDAT can largely reduce computing consumption in decoding, showing its potential for real-time applications.

Model	Pre-trained?	PersonaChat				ROCStory			Latency	Throughput
		B-1	B-2	D-1	D-2	B-1	B-2	D-4	ms/sample	samples/s
Autoregressive Text Generation Models										
Transformer	N	18.37	8.07	1.43	10.04	30.68	14.67	35.18	168 (1.1x)	28 (1.1x)
MASS	Y	26.82	14.70	1.20	7.58	35.02	16.96	51.20	180 (1.0x)	25 (1.0x)
w/ Sampling	Y	23.90	12.13	1.85	13.09	32.56	14.97	73.72	130 (1.4x)	77 (3.0x)
BART	Y	26.84	14.69	1.39	8.85	35.45	17.22	49.03	199 (0.9x)	23 (0.9x)
w/ Sampling	Y	24.00	12.31	1.97	14.50	33.95	15.28	73.62	143 (1.3x)	69 (2.7x)
Non-autoregressive Text Generation Models										
Vanilla NAT	N	18.33	6.37	0.43	0.96	28.44	11.29	89.13	23 (7.8x)	**703 (27.7x)**
BANG	Y	17.38	7.33	**2.12**	**23.02**	29.38	11.78	**92.10**	**18 (10.1x)**	649 (25.6x)
MIST	Y	18.55	8.86	0.54	2.56	23.57	9.09	8.15	25 (7.3x)	330 (13.0x)
PreDAT (Ours)	Y	27.06	15.05	1.33	8.31	34.11	17.17	57.50	24 (7.6x)	507 (20.0x)
w/ Sampling	Y	24.23	12.29	1.77	15.62	32.52	15.61	74.37	24 (7.4x)	514 (20.3x)
w/ BeamSearch	Y	**27.31**	**15.39**	1.15	6.30	**34.61**	**17.84**	50.55	48 (3.7x)	318 (12.6x)
w/o Pre-training	N	21.96	10.38	0.52	3.29	31.81	15.41	52.97	25 (7.2x)	562 (22.2x)

Table 3: Performance on open-ended text generation datasets. **Latency** and **Throughput** are evaluated on the test set of PersonaChat. Average scores are not shown because they cannot reflect the trade-off between quality and diversity. We utilize corpus BLEU on all datasets, whose values may be different from some previous results utilizing sentence BLEU (Liu et al., 2021). The metrics include BLEU-1/2 (B-1/2) and Distinct-1/2/4 (D-1/2/4).

Open-Ended Text Generation We further test PreDAT on two open-ended text generation tasks, dialog generation and story generation. Open-ended text generation tasks encourage the model to produce novel and diverse outputs, where sampling decoding methods are commonly adopted to promote generation diversity.

Therefore, in addition to lookahead decoding and beamsearch, we also introduce Nucleus sampling (Holtzman et al., 2020). Specifically, we set $p = 0.9$ and the temperature $\tau = 1$ for PreDAT. For MASS and BART, we also use $p = 0.9$, but $\tau = 0.8$ on PersonaChat and $\tau = 0.7$ on ROCStory to achieve similar diversity as PreDAT.

We present the evaluation results in Table 3 and the trade-off of quality and diversity by tuning the temperature in Figure 4. Generally, the comparison of quality metrics is similar to closed-ended generation: PreDAT largely outperforms NAR baselines and achieves comparable BLEU scores to AR models. Moreover, we highlight two findings:

- PreDAT generates plausible outputs in open-ended tasks while previous NAR models cannot. Open-ended generation tasks usually have targets with diverse expressions, which worsens the multi-modality problem and seriously degrades the NAR generation quality. Specifically, MIST shows very low diversity because it generates numerous repetitions, and BANG shows very high diversity because it introduces many incomprehensible n-grams. In contrast, PreDAT has a reasonable quality-diversity trade-off, showing its ability to address the serious challenges brought by the multi-modality problem.

- PreDAT achieves a flexible quality and diversity trade-off. As shown in Figure 4, PreDAT is slightly better than two AR baselines

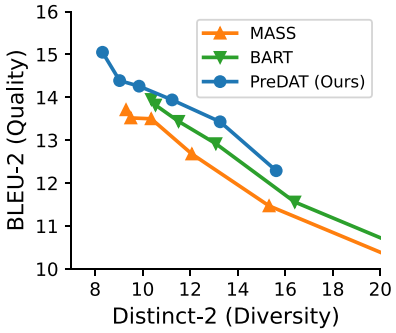

Figure 4: Trade-off curves of quality and diversity on PersonaChat. All models use Nucleus sampling with $p = 0.9$ and temperature τ from $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$.

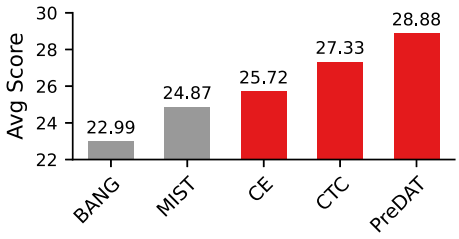

Figure 5: Average performance of previous baselines (Gray) and NAR models pre-trained by our proposed task with different loss functions (Red). The shown scores are the average of automatic metrics on XSUM.

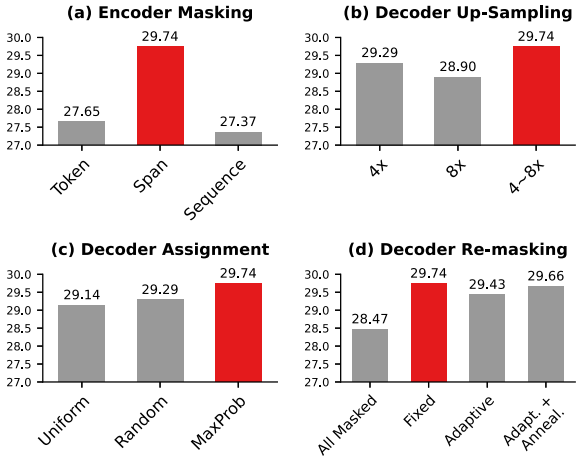

Figure 6: Comparisons of pre-training strategies by the average validation score on SQuAD1.1 and XSUM. All models are pre-trained for 100k steps for energy saving. The strategies in our final model are marked in Red.

w.r.t. the trade-off curves by tuning the decoding temperature. It demonstrates that PreDAT can meet the diversity requirement of open-ended text generation, verifying its generality in text generation.

5.4 Ablation Study

In this section, we conduct ablation studies to reveal how our designs contribute to the results.

Loss Function In PreDAT, we utilize the DAT loss to alleviate the multi-modality problem, which plays an important role in the pre-training. Notably, our pre-training task can be combined with other NAR losses, so we compare the DAT loss against CTC (Graves et al., 2006; Libovický and Helcl, 2018) and the token-level cross-entropy loss (CE).

Specifically, the same loss function is applied in both pre-training and fine-tuning to avoid discrepancies between the two training stages. For CTC, we randomly sample the up-sampling ratio from $[1.5, 2]$. For CE, we do not use up-sampling (i.e., $\lambda = 1$) because the CE loss requires an output sequence with the same length as the target.

As shown in Figure 5, we find: (1) It is important to incorporate alignment-based NAR losses in pre-training, where CTC and DAT losses bring substantial improvements compared with the CE loss. (2) The NAR model pre-trained with CE still outperforms previous pre-trained NAR baselines, verifying the effectiveness of our pre-training task in preserving sentence-level consistency and promoting bidirectional dependencies.

Pre-training Strategy Our proposed pre-training task includes several strategies for constructing the training data pair. To evaluate the effects of these strategies, we design four groups of comparisons as follows, whose results are shown in Figure 6.

(a) Stage 1: Encoder Masking. Besides **Span** masking, we use two other strategies including **Token** masking that independently samples masked positions (Devlin et al., 2019), and **Sequence** masking that masks a single consecutive sequence. All strategies mask the same ratio of tokens. We conclude that the masked spans should not be too short (about 1∼3 tokens in token masking) or too long (about 90 tokens in sequence masking), which prevents the NAR model from learning prediction consistency or make the prediction too difficult.

(b) Stage 2, Step 1: Up-sample Ratios. We compare the random sampling ratio (**4∼8x**) against fixed up-sampling ratios (**4x** and **8x**). We find that random up-sampling can diversify the DAG structure, which works as a data augmentation method and thus benefits the downstream performance.

(c) Stage 2, Step 2: Assignment Strategies. Besides the proposed assignment strategy according to the path probability (**MaxProb**), we use **Uniform** and **Random** assignment that assigns the target into the decoder input uniformly or randomly. We find the MaxProb assignment can better determine the lengths of each masked segment according to the model's own prediction, leading to slightly better results than the other strategies.

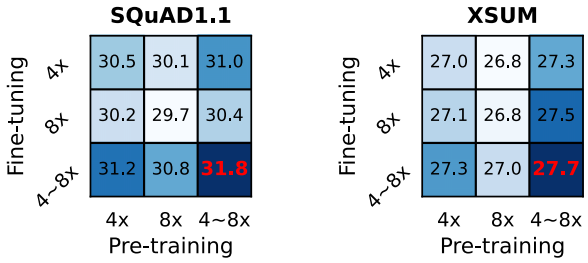

Figure 7: Validation performance under various combinations of up-sampling strategies in pre-training and fine-tuning. The shown score is the average of automatic metrics. 4x and 8x indicates fixed up-sampling ratios, and 4∼8x indicates random ratios sampled from $[4, 8]$. All models are pre-trained for only 100k steps.

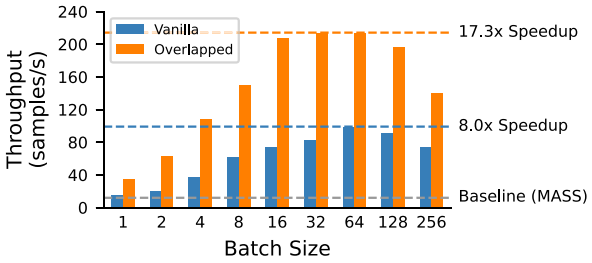

Figure 8: Throughput speedups with the vanilla and overlapped decoding on the test set of XSUM.

(d) Stage 2, Step 3: Re-masking Strategies. Besides the **Fixed** masking strategy, we also try **Adaptive** and **Adaptive + Annealing** masking strategies proposed by Qian et al. (2021), where they adjust the masking ratio by the difficulties of the sample. It shows that these strategies have similar performance, outperforming the fully masked decoder input (**All Masked**), which verifies the importance of introducing information in the decoder input for bidirectional dependency modelling. However, the adaptive masking is less effective in pre-training, so we use the fixed masking ratio for simplicity.

Up-sampling Ratio in Fine-tuning As described in Section 4.2, we obtain the decoder input length in fine-tuning by up-sampling. To investigate how the up-sampling strategies affect performance, we evaluate different combinations of up-sampling ratios in pre-training and fine-tuning.

As shown in Figure 7, random up-sampling always benefits the performance in pre-training and fine-tuning, together bringing an improvement of about 1.2 scores. It indicates that varying the DAG size is an important trick in training PreDAT. Moreover, the up-sampling ratios in pre-training and fine-tuning do not need to be the same, which can be helpful if smaller DAG sizes are preferred in downstream tasks due to limited memory budget.

Overlapped Decoding Overlapped decoding aims to improve the decoding throughput by overlapping the execution of DAG prediction and beam search decoding. To verify its effectiveness, we evaluate the speedup with various batch sizes on XSUM.

As shown in Figure 8, our overlapped decoding brings a 17.3x speedup with a batch size of 32, largely outperforming the vanilla one. We also note that throughput starts to decline as batch size increases, possibly because the introduced paddings increase the consumption of invalid computations.

5.5 Analysis

In this section, we investigate the reasons why PreDAT achieves better automatic scores than pre-trained AR baselines, which may provide some insights for future NAR generation studies.

PreDAT Alleviates Error Accumulation. Error accumulation is a major concern of autoregressive generation (Bengio et al., 2015; Ranzato et al., 2016; Arora et al., 2022), where a prediction error may be propagated into later decoding steps, leading to low-quality generated sentences. In contrast, NAR models naturally avoid the problem due to their unbiased prediction order.

To verify that PreDAT has advantages in tackling error accumulation, we compare PreDAT against two AR models with different decoding orders, a left-to-right (L2R) one and a right-to-left (R2L) one. Specifically, we fine-tune MASS on the downstream datasets using the two generation orders. We find that MASS still performs well in right-to-left decoding, with a performance drop of less than 0.5 scores. Then we calculate the average token prediction accuracy bucketed by the relative position, formally defined as

$$\mathrm{Acc}(D) = \mathrm{Average}(\mathbb{I}(\hat{Y}_j^{(i)} \in Y^{(i)}))$$
$$\text{for } 1 \le i \le N, 1 \le j \le |\hat{Y}^{(i)}|, \frac{j}{|\hat{Y}^{(i)}|+1} \in D,$$

where $\mathrm{Acc}(D)$ is the average prediction accuracy on the interval D, $Y^{(i)}$ is the i-th sample in

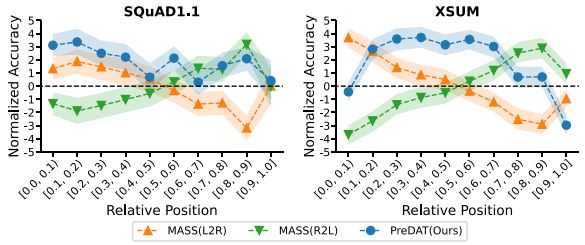

Figure 9: Normalized prediction accuracy $\Delta\text{Acc}(D)$ bucketed by relative positions. The shaded area is 95% confidence interval by bootstrap sampling (Efron and Tibshirani, 1994). L2R: left-to-right, R2L: right-to-left.

Dataset	Model	Knowledge			PARENT-T		
		P	R	F1	P	R	F1
XSUM	MASS	35.1	9.7	14.7	35.1	8.5	13.1
	PreDAT	**36.3**	**9.9**	**14.9**	**36.4**	**8.6**	**13.3**
PersonaChat	MASS	19.6	17.2	17.8	13.2	**11.3**	**11.5**
	PreDAT	**21.1**	**17.7**	**18.5**	**13.8**	11.0	**11.5**

Table 4: Relevance to the input on XSUM and PersonaChat. We utilize two automatic metrics, Knowledge F1 and PARENT-T. P: Precision, R: Recall.

the test set, $\hat{Y}^{(i)}$ is the i-th model outputs, and j indicates the position. Moreover, since prediction difficulties vary with the positions (e.g., the last token is always punctuation), we utilize a normalized accuracy:

$$\Delta\text{Acc}(D) = \text{Acc}(D) - \frac{\text{Acc}_{\text{L2R}}(D) + \text{Acc}_{\text{R2L}}(D)}{2},$$

where $\text{Acc}_{\text{L2R}}(D)$ and $\text{Acc}_{\text{R2L}}(D)$ indicate the prediction accuracy of L2R and R2L MASS.

As shown in Figure 9, we find that MASS has a strong tendency to predict earlier generated tokens more accurately than later generated ones, which applies to both left-to-right and right-to-left models. In contrast, our PreDAT shows no significant preference for any positions because it predicts all tokens simultaneously, which reveals the advantages of unbiased prediction order in NAR generation models.

PreDAT Improves the Relevance to the Input. Previous studies empirically found that AR generated texts may lose relevance to the input sentences, which is also known as hallucination (Maynez et al., 2020; Ji et al., 2022) or off-prompt errors (Dou et al., 2022). One explanation is that AR models may be distracted by its generated prefixes, which can be avoided in NAR generation (Huang et al., 2021).

To verify our hypothesis, we introduce two metrics to evaluate the relevance to inputs: Knowledge F1 (Shuster et al., 2021) and PARENT-T (Dhingra et al., 2019; Wang et al., 2020). Knowledge F1 measures the unigram F1 between generated sentences and the input knowledge, while PARENT-T measures n-gram entailment. Both metrics require the extraction of knowledge pieces that should appear in the generated sentences. For simplicity, we take each sentence in the passage (of XSUM) or the persona profile (of PersonaChat) as a piece of knowledge and further filter out the stop words.

As shown in Table 4, PreDAT achieves better precision on both datasets in using the input knowledge compared with MASS (+1.2 on average). It indicates that PreDAT is less likely to produce irrelevant keywords, justifying our hypothesis that the NAR model can better concentrate on the input. However, we also notice that PreDAT and MASS have comparable performance on recall, showing that it is still challenging to cover more keywords.

5.6 Manual Evaluation

Although PreDAT shows surprising performance in automatic evaluation, it is still questionable whether these automatic metrics are reliable when comparing AR and NAR models. In this section, we conduct a manual evaluation that compares PreDAT against pre-trained AR and NAR baselines.

Settings We compare PreDAT against three baselines, two NAR models (BANG and MIST) and an AR model (MASS). We randomly selected 150 samples in SQuAD1.1, accounting for 600 generated sentences for the four models. For each sample, three annotators were asked to rank the outputs from two dimensions: **grammaticality** measures whether the output contains any grammatical errors, and **appropriateness** measures whether the output is reasonable for the given context.

Results The results are shown in Table 5, where we highlight two findings: (1) PreDAT achieves a significant quality improvement over previous

	Grammaticality				Appropriateness			
	Win	Tie	Lose	κ	Win	Tie	Lose	κ
Comparison against Non-autoregressive Models								
vs. BANG	75.3**	12.0	12.7	0.66	69.8**	17.3	12.9	0.59
vs. MIST	66.7**	18.0	15.3	0.50	57.1**	26.0	16.9	0.47
Comparison against Autoregressive Models								
vs. MASS	15.1	47.8	37.1**	0.32	32.2	36.7	31.1	0.46

Table 5: Manual evaluation results on SQuAD1.1. Fleiss' κ is shown for iter-rater reliability (all are fair agreement or above). * and ** indicate p-value $<$ 0.05 and 0.01 in the sign test, respectively.

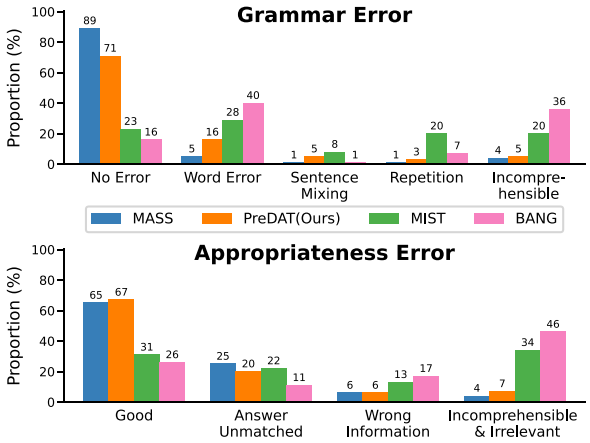

Figure 10: Proportion of samples with different error types in terms of grammaticality and appropriateness on SQuAD1.1. *Word Error*: containing less than two wrong/missing/redundant tokens. *Sentence Mixing*: can be splitted into two (nearly) grammatically correct sentences with a shared fragment. *Answer Unmatched*: the generated question is not matched with the given answer. *Wrong Information*: using incorrect or unmentioned information in the passage.

NAR models, where annotators highly prefer PreDAT (with Win% + Tie% $>$ 83%). (2) There is still a quality gap between PreDAT and the AR model. Although PreDAT achieves higher word overlap in automatic evaluation, it exhibits poorer grammaticality in human ratings. A possible reason is that PreDAT preserves better relevance to the inputs, leading to the higher word overlap, however, is still weaker than AR models in preserving the consistency among generated tokens.

Typical Errors and Case Study To better understand how PreDAT makes errors, we investigate the typical errors in the generated outputs. Specifically, we randomly chose 100 samples from SQuAD1.1, collected the outputs of the four models, and then manually annotated the errors in these outputs.

Figure 10 presents the proportions of error types. In terms of grammaticality, we find that PreDAT addresses the major problems in previous NAR models, such as incomprehensible outputs and repetitions, well. However, there are still some word errors, which affect only a small fragment of the sentence but are very obvious to human readers, leading to the unsatisfying result. We believe the problem can be alleviated by post-editing or iterative refinement, which we leave for future work. In terms of appropriateness, PreDAT has comparable performance to MASS in error distributions, showing its ability to extract and organize information to form appropriate outputs.

To support the above discussions, we show some output cases in Table 6. We find that previous NAR models usually generate low-quality texts, whereas PreDAT achieves significant improvement. Moreover, PreDAT maintains a strong relevance to the inputs, yet it can occasionally introduce grammatical errors. In contrast, MASS generates plausible outputs, but they may not always be faithful. This observation highlights the distinctive behaviors between AR and NAR models.

6 Limitations

Although PreDAT achieves a significant advancement in NAR generation, it still faces the following limitations:

(1) Although PreDAT achieves superior performance in automatic evaluation, it still significantly underperforms AR models in grammaticality according to human evaluation (as discussed in Section 5.6). This inconsistency can be attributed to the different biases of AR and NAR models: AR models tend to generate fluent outputs but may sacrifice relevance to the input, while NAR models prioritize relevance but may incur grammatical errors. It is important to take the behavior into consideration when applying PreDAT to real-world applications.

(2) PreDAT may struggle with capturing long-range coherence, because NAR models are inherently weak in modeling token dependencies, and PreDAT is pre-trained only on predicting 15-token-long fragments. Notably, our experiments

	SQuAD1.1
Passage:	(74 words omitted) . . . JFK and Newark Liberty were the busiest and fourth busiest U.S. gateways for international air passengers, respectively, in 2012 . . . (72 words omitted)
Answer:	Newark Liberty International Airport
BANG	what airport *in busiest airport in the u* .
MIST	what is john f . kennedy international airport *john f busiest international airport and laguardia* ?
MASS	what is the name of *the busiest airport in new york* ?
PreDAT (Ours)	what is the name of the fourth busiest airport for international air passengers in 2012 ?
Passage:	(102 words omitted) . . . The FDIC guarantees the funds of all insured accounts up to US $ 100, 000 . . . (72 words omitted)
Answer:	US $ 100, 000
BANG	*what* much the deposits of *deposits of allmac deposits ?*
MIST	what is the funds of all insured *ins allmac accounts to ?*
MASS	how much does the fdic guarantee the funds of all insured accounts ?
PreDAT (Ours)	how much *is* the fdic guarantee the funds of all insured accounts ?
Passage:	When one Republican presidential candidate for the 2016 election ridiculed the liberalism of "New York values" in January 2016, Donald Trump, leading in the polls, vigorously defended his city . . . (68 words omitted)
Answer:	Donald Trump
BANG	who *did* the republican *the against new values " in*
MIST	who was the leader *the "* new york in 2016 ?
MASS	who was *the republican presidential candidate* for 2016 ?
PreDAT (Ours)	who *led* the polls in 2016 ?
	PersonaChat
Persona:	(9 words omitted) . . . I like to listen to country music . . . (24 words omitted)
History:	A: Hello I like to travel. B: Hello, how are you tonight? I do too and love to cook. A: I would love to see europe.
BANG	i would like *to too* but i am a *to* .
MIST	what do you do for a living . *for a living ?*
MASS	i have never been to europe , but i have always wanted to go to *australia* .
PreDAT (Ours)	i would love to go to europe . i am listening to country music .
Persona:	I am an eccentric hair stylist for dogs . . . (27 words omitted)
Dialog History:	(24 tokens omitted) ... A: I am doing wonderful, now that I avoided the mangoes. I am allergic. B: Oh sorry to hear that I like going out with my friends.
BANG	i *do you like.*
MIST	what do you do for a living *for a living ?*
MASS	do you have any pets ? *i have a dog* .
PreDAT (Ours)	what do you like to do *with* fun ? i am a hair stylist .
Persona:	. . . (43 words omitted)
Dialog History:	(131 tokens omitted) ... B: I bet you can learn a lot studying ice, must be cold though. A: It is. Some people freeze to death. B: Yikes, too cold for me. i will stay home with my pets!
BANG	i *do you do any*
MIST	what do you do *. pets . how . you ?*
MASS	do you have *any hobbies besides music* ?
PreDAT (Ours)	what kind of pets do you *do* ?

Table 6: Cases of model outputs on SQuAD1.1 and PersonaChat. Grammatical errors are marked in *red*. The phrases that are faithful to the input are marked in blue, whereas the unfaithful ones are marked in *brown*. All generated sentences are in lowercase.

are conducted on relatively short text generation (whose length statistics are shown in Table 1), and the performance on longer text generation tasks requires further investigation.

(3) Compared with AR models, PreDAT requires more GPU memory during inference and takes more time in fine-tuning (typically 2∼4 times in our experiments). This is because PreDAT's decoder has to process a much longer sequence.

7 Conclusion

In this paper, we propose a pre-training task to promote sentence-level consistency and bidirectional dependencies for NAR generation. We

demonstrate that combining the state-of-the-art NAR models with appropriate pre-training can lead to efficient and high-quality text generation on a wide range of tasks, where our PreDAT largely outperforms previous NAR pre-trained models in generation quality. We further show that, compared with AR models, PreDAT alleviates error accumulation and enhances relevance to inputs, but still introduces non-negligible grammatical problems, thereby providing new insights into the strengths and weaknesses of NAR generation.

Acknowledgments

This paper was supported by the National Science Foundation for Distinguished Young Scholars (with grant no. 62125604) and the Guoqiang Institute of Tsinghua University, with grant no. 2020GQG0005. We are grateful to the action editor and the anonymous reviewers for their valuable suggestions and feedback.

References

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. 2022. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22–27, 2022*, pages 700–710. https://doi.org/10.18653/v1/2022.findings-acl.58

Yu Bao, Hao Zhou, Shujian Huang, Dongqi Wang, Lihua Qian, Xinyu Dai, Jiajun Chen, and Lei Li. 2022. latent-GLAT: Glancing at latent variables for parallel text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22–27, 2022*, pages 8398–8409. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.575

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7–12, 2015, Montreal, Quebec, Canada*, pages 1171–1179.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Bhuwan Dhingra, Manaal Faruqui, Ankur P. Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 – August 2, 2019, Volume 1: Long Papers*, pages 4884–4895. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1483

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021a. Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021, pages 3431–3441. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.266

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021b. Understanding and improving lexical choice in non-autoregressive translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net.

Liang Ding, Longyue Wang, Shuming Shi, Dacheng Tao, and Zhaopeng Tu. 2022. Redistributing low-frequency words: Making the most of monolingual data in non-autoregressive translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22–27, 2022*, pages 2417–2426. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.172

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2022. Is GPT-3 text indistinguishable from human text? Scarecrow: A framework for scrutinizing machine text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22–27, 2022*, pages 7250–7274. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.501

Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2849–2859. PMLR.

Bradley Efron and Robert Tibshirani. 1994. *An Introduction to the Bootstrap*, Chapman and Hall/CRC. https://doi.org/10.1201/9780429246593

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13–18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3515–3523. PMLR.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 6111–6120. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1633

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25–29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM. https://doi.org/10.1145/1143844.1143891

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 – May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu and Xiang Kong. 2021. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1–6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 120–133. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.11

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019,*

December 8–14, 2019, Vancouver, BC, Canada, pages 11179–11189.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108. https://doi.org/10.1162/tacl_a_00302

Junliang Guo, Linli Xu, and Enhong Chen. 2020a. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 376–385. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.36

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020b. Incorporating BERT into parallel sequence decoding with adapters. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net.

Chenyang Huang, Hao Zhou, Osmar R. Zaïane, Lili Mou, and Lei Li. 2022a. Non-autoregressive translation with layer-wise prediction and deep supervision. *The Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*. https://doi.org/10.48550/arXiv.2110.07515

Fei Huang, Zikai Chen, Chen Henry Wu, Qihan Guo, Xiaoyan Zhu, and Minlie Huang. 2021. NAST: A non-autoregressive generator with word alignment for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1–6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1577–1590. https://doi.org/10.18653/v1/2021.findings-acl.138

Fei Huang, Tianhua Tao, Hao Zhou, Lei Li, and Minlie Huang. 2022b. On the learning of non-autoregressive transformers. In *International Conference on Machine Learning, ICML 2022, 17–23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 9356–9376. PMLR.

Fei Huang, Hao Zhou, Yang Liu, Hang Li, and Minlie Huang. 2022c. Directed acyclic transformer for non-autoregressive machine translation. In *International Conference on Machine Learning, ICML 2022, 17–23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 9410–9428. PMLR.

Xiao Shi Huang, Felipe Pérez, and Maksims Volkovs. 2022d. Improving non-autoregressive translation models without distillation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25–29, 2022*. OpenReview.net.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *CoRR*, abs/2202.03629. https://doi.org/10.1145/3571730

Ting Jiang, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Liangjie Zhang, and Qi Zhang. 2021. Improving non-autoregressive generation with mixup training. *arXiv preprint*, abs/2110.11115v1. https://doi.org/10.48550/arXiv.2110.11115

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016*, pages 1317–1327. The Association for Computational Linguistics. http://doi.org/10.18653/v1/D16-1139

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*. https://doi.org/10.48550/arXiv.1412.6980

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive

neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 – November 4, 2018*, pages 1173–1182. Association for Computational Linguistics. http://doi.org/10.18653/v1/D18-1149

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 7871–7880. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.703

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2022. ELMER: A non-autoregressive pre-trained language model for efficient and effective text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7–11, 2022*, pages 1044–1058. Association for Computational Linguistics.

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 5707–5712. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1573

Jindrich Libovický and Jindrich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 – November 4, 2018*, pages 3016–3021. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1336

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, Pengcheng Wang, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, Ruofei Zhang, Winnie Wu, Ming Zhou, and Nan Duan. 2021. GLGE: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1–6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 408–420. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.36

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 1906–1919. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.173

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016*, pages 839–849. Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1098

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 – November 4, 2018*, pages 1797–1807. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1206

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Demonstrations, pages 48–53. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-4009

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6–12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL. https://doi.org/10.3115/1073083.1073135

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 – May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Weizhen Qi, Yeyun Gong, Jian Jiao, Yu Yan, Weizhu Chen, Dayiheng Liu, Kewen Tang, Houqiang Li, Jiusheng Chen, Ruofei Zhang, Ming Zhou, and Nan Duan. 2021. BANG: Bridging autoregressive and non-autoregressive generation with large scale pretraining. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8630–8639. PMLR.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future *n*-gram for sequence-to-sequence pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16–20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2401–2410. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.217

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 1993–2003. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.155

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016*, pages 2383–2392. Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1264

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings*. https://doi.org/10.48550/arXiv.1511.06732

Chenze Shao, Xuanfu Wu, and Yang Feng. 2022. One reference is not enough: Diverse distillation with reference selection for non-autoregressive translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10–15, 2022*, pages 3779–3791. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.277

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval

augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16–20 November, 2021*, pages 3784–3803. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-emnlp.320

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Yixuan Su, Deng Cai, Yan Wang, David Vandyke, Simon Baker, Piji Li, and Nigel Collier. 2021. Non-autoregressive text generation with pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 – 23, 2021*, pages 234–243. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.18

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhi-Hong Deng. 2019. Fast structured decoding for sequence models. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada*, pages 3011–3020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xiaohui Wang, Yang Wei, Ying Xiong, Guyue Huang, Xian Qian, Yufei Ding, Mingxuan Wang, and Lei Li. 2022. Lightseq2: Accelerated training for transformer-based models on GPUs. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis, Dallas, TX, USA, November 13–18, 2022*, pages 1–14. IEEE. https://doi.org/10.1109/SC41404.2022.00043

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 1072–1086. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.101

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *arXiv preprint*, abs/1910.03771v5. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1205

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7–13, 2015*, pages 19–27. IEEE Computer Society. https://doi.org/10.1109/ICCV.2015.11