UT5: Pretraining Non autoregressive T5 with unrolled denoising

Mahmoud G. Salem, Jiayu Ye, Chu-Cheng Lin, Frederick Liu
Google
{* }@google.com

Abstract

Recent advances in Transformer-based Large Language Models have made great strides in natural language generation. However, to decode K tokens, an autoregressive model needs K sequential forward passes, which may be a performance bottleneck for large language models. Many non-autoregressive (NAR) research are aiming to address this sequentiality bottleneck, albeit many have focused on a dedicated architecture in supervised benchmarks. In this work, we studied unsupervised pretraining for non auto-regressive T5 models via unrolled denoising and shown its SoTA results in downstream generation tasks such as SQuAD question generation and XSum.

1 Introduction

Large language models (LLMs) have revolutionized natural language processing (NLP) by enabling automatic text generation and prediction. Traditionally, language models are *autoregressive*: they generate a sequence of tokens one by one, conditioning each token on the previously generated ones. While this approach has led to impressive results (OpenAI, 2023; Anil et al., 2023), it suffers from slow inference due to its sequential nature. Several studies (Gu et al., 2018; Ghazvininejad et al., 2019a) have explored the use of non-autoregressive generation for language modeling, where tokens can be generated in parallel, without the need of conditioning on previously generated ones. Non-autoregressive generation has shown promising results in terms of efficiency and speed, and has the potential to be applied to various NLP tasks (Liu et al., 2020). Pretraining has proven the foundational procedure for autoregressive generation (Devlin et al., 2019; Radford et al., 2018). However, few studies have focused on pretraining for non-autoregressive language modeling for efficient language generation . The main advantage of non-autoregressive generation is parallel generation of all tokens, making it faster than auto-regressive generation. However, non-autoregressive generation usually exhibits quality gaps when comparing with similar sized autoregressive models (Gu and Kong, 2020).

In this paper, we propose a pretraining regime to improve the quality of non-autoregressive generation. To explore the effects of pretraining on decoder-only models, we employed step-unrolled denoising (Savinov et al., 2021) to pretrain the models. In the rest of the paper, we describe our proposed pretraining regime in detail and evaluate its effectiveness in improving the quality of efficientnon-autoregressive text generation.

Our contributions are:

- Introduce training regime for non-autoregressive models for efficient language generation,
- We show that the non-autoregressive pretraining with unrolled denoising significantly improves the results on downstream benchmarks compared to training from scratch.
- We are able to achieve SoTA results on downstream evaluations with similar parameter count.

2 Related work

Pretraining language models on large-scale data has shown great success for auto-regressive language models (Devlin et al., 2018; Ilić et al., 2018; Radford et al., 2018). The models are pre-trained on large-scale data in a self-supervised manner then finetuned on downstream tasks like text classification and machine translation. While pre-training is a standard in many autoregressive language tasks, it is understudied in non-autoregressive settings. Some efforts have been made to study and adapt pre-training for non auto-regressive models. (Guo et al., 2020) incorporates two BERT models into

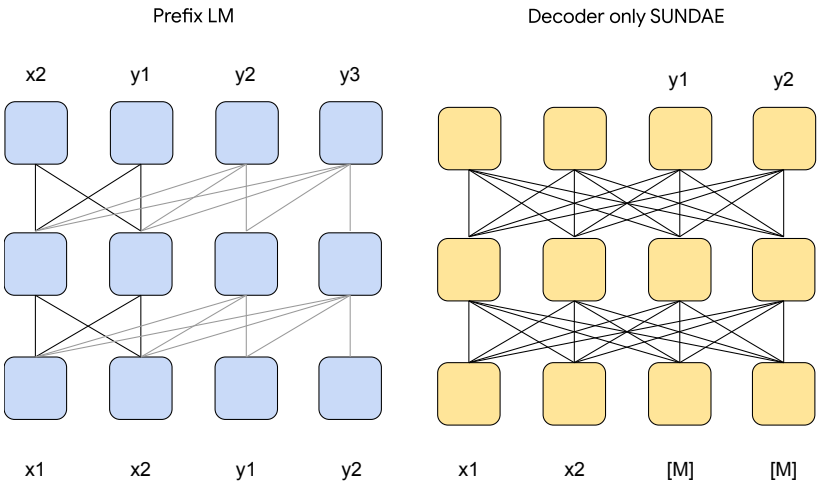

Figure 1: Illustration of prefix Language Model versus Decoder-only bidirectional de-noising model.

machine translation using mask-predict decoding method, their method utilizes two pre-trained BERT models one as the encoder and one as the decoder, and then inserts adapter layers into each layer. (Su et al., 2021) follows similar regime but uses one BERT as the backbone model and then add a CRF output layer which captures the target side dependency and improves the performance. Further (Li et al., 2022) introduced CeMAT which uses a bidirectional encoder and decoder architecture. The model is jointly trained with Masked Language modeling (MLM) for the decoder and Conditional Masked Language Modeling (CMLM) for the decoder with a cross attention module for bridging them. The model seeks to enhance multilingual ability in machine translation by pre-training on large-scale monolingual and bilingual texts in many languages and using an aligned code-switching strategy than finetuned on NAT and AT tasks.

SUNDAE (Savinov et al., 2021) is a novel method for training denoising models for text generation. SUNDAE improves upon traditional denoising autoencoders by unrolling the decoding process for multiple steps and adding noise at each step. resulting in a more robust and effective model for generating text. The authors demonstrated the effectiveness of the SUNDAE method in several text generation tasks, including sentence completion and language modeling, and showed that it outperformed other state-of-the-art methods in terms of both quality and efficiency. The SUNDAE method provides a promising approach to text generation and has practical applications in various natural language processing tasks. However, SUNDAE language generation suffers a huge drop in performance when adapted in non-auto-regressive generation setting. In this study we focus on recovering the drop in performance using large-scale pretraining.

BANG (Qi et al., 2021) investigated pretraining an LLM using a mixture of autoregressive and non-autoregressive objective functions. Their downstream tasks include machine translation, summarization, and dialogue generation. BANG achieves state-of-the-art performance on several benchmark datasets, demonstrating the effectiveness of large-scale pretraining for bridging the gap between autoregressive and non-autoregressive language generation. We consider the BANG model to be a potential baseline, where the non-autoregressive parametrization simply dropped conditioning on previously generated tokens.

3 Method

Pretraining techniques such as masked language modeling (MLM) on large-scale data have shown to be effective in improving the performance of neural language models. In this section, we investigate the effects of large-scale pretraining on decoder-only non-autoregressive models. We adopted SUNDAE (Savinov et al., 2021), a two-step training method for generative modeling of discrete sequences using denoising autoencoders and Markov chain models. The training process includes unrolled denoising, which involves starting the chain from corrupted data samples instead of the prior distribution. The model learns to denoise samples that it is likely to

encounter during full unrolling used at sample time.

$$L^{(t)}(\theta) := -\mathbb{E}_{\substack{\mathbf{x}\sim p_{\text{data}} \\ \mathbf{x}_0\sim q(\cdot|\mathbf{x}) \\ \mathbf{x}_1\sim f_\theta(\cdot|\mathbf{x}_0)}}[\log f_\theta(\mathbf{x}|\mathbf{x}_i)], \quad (1)$$

where $\mathbf{x}_i$ is the ith iteration denoised result, $q(\cdot|\mathbf{x})$ is the corruption function, and f_θ is the network.

We investigate the effect of pretraining on the decoder-only architecture proposed in (Radford et al., 2018) combined with SUNDAE two-step training procedure as our baseline model. The pretraining is done on the Colossal Clean Crawled Corpus (C4) dataset. The pretraining objective is similar to prefix language modeling but with bidirectional attention as shown in Figure 1. Following pretraining, we finetune the model on several downstream tasks.

3.1 Model Details

We ground the work on T5 base (Raffel et al., 2020) and develop a decoder-only model on top. Our baseline model utilizes a decoder-only transformer-based architecture with bidirectional self-attention. Specifically, we employ a 12-layer decoder with hidden states of dimension 768. This is comparable with BANG with 6 layers of encoder and 6 layers of decoder with the same hidden dimension.

Several NAR techniques (Gu et al., 2018; Savinov et al., 2021) try to incorporate the output sentence length information during the training allowing NAR models to have some approximate of the output length. To keep our study simple and focused on the value of pretraining, we omit the use of length prediction neither as an auxiliary loss or a separate module. Alternatively, the model is trained to predict padding tokens to fill the target sequence buffer.

3.2 Training Strategy

During the pretraining phase, our model underwent training for 1 million steps on the C4 dataset with a batch size of 128 and a sequence length of 512 inputs and 114 targets. We explore span corruption and prefix LM strategies during pretraining while observing the latter is more stable. One of the hypothesis is a single span corruption target is shorter hence less meaningful to unroll. Hence for the studies below, we use Prefix LM objective with bidirectional attention (Figure. 1). This process allowed the model to develop a comprehensive understanding of language patterns and contextual relationships.

For the subsequent finetuning stage, the model is fine-tuned on a specific downstream task for 50k steps, employing a learning rate of 0.0001. The pretraing helps the model to efficiently finetune on different downstream tasks with fewer number of steps. The finetuning process further refined the model's parameters and enabled it to adapt to the nuances and requirements of the target task. During the model inference evaluation, the model unrolls 10 steps from the mask then decodes text as output.

4 Experiments

We conduct the experiments to study the effect of pretraining on decoder-only NAR models. We analyze the performance on these models on downstream tasks with and without pretraining. Our experiments are all conducted through JAX/Flax (Bradbury et al., 2018) using the T5x framework (Roberts et al., 2022). We use TPU-v3 chips for pretraining and finetuning, typical pretraining jobs use 256 chips for a week and finetuning jobs use 16 to 64 chips for a day.

4.1 Datasets

Pretraining. For our pretraining experiments, we use the C4 dataset, which is a large-scale web document corpus created by scraping the Common Crawl data. The C4 dataset contains over 750GB of text data and includes a diverse range of topics, such as news, blogs, and online forums. The text data in the C4 dataset is preprocessed and tokenized into individual sentences, making it suitable for language modeling tasks. The C4 dataset has several advantages over other datasets for pretraining, such as its large size and diversity. The size of the dataset allows for the training of large-scale language models, which have been shown to achieve state-of-the-art performance on various NLP tasks. Additionally, the diversity of the C4 dataset helps to capture the different styles and registers of language used in the web documents, making the pretraining models more robust to different text domains.

To evaluate our approach, we conduct experiments on following two popular generation benchmarks for downstream evaluation:

XSum. The XSum dataset (Narayan et al., 2018) contains over 227,000 news articles and their corresponding summaries from the British Broad- casting Corporation (BBC). The articles are taken from a wide range of topics, such as politics, business, sports, and entertainment. The summaries are writ-

Model	Pretrain	XSum				SQuAD	
		ROUGE-1	ROUGE-2	ROUGE-L	OVERALL	ROUGE-L	BLEU-4
NAT (Gu et al., 2018)	No	24.04	3.88	20.32	16.08	31.51	2.46
iNAT (Lee et al., 2018)	No	24.02	3.99	20.36	16.12	32.44	2.33
CMLM (Ghazvininejad et al., 2019b)	No	23.82	3.60	20.15	15.86	31.58	2.51
LevT (Gu et al., 2019)	No	24.75	4.18	20.87	16.60	31.38	2.27
BANG NAR (Qi et al., 2021)	Yes	32.59	8.98	27.41	22.99	44.07	12.75
BANG semi-NAR	Yes	34.71	11.71	29.16	25.19	**47.39**	**17.62**
Ours (no prefix-lm pretraining)	No	32.56	11.8	26.17	23.51	31.36	3.903
Ours (with prefix-lm pretraining)	Yes	**35.80**	**14.03**	**29.27**	**26.36**	45.75	12.47

Table 1: NAR results on the XSum and SQuAD 1.1 question generation.

ten to capture the main idea and salient points of the articles in a single sentence. The average input and output lengths are 358.5 and 21.1, respectively.

SQuAD 1.1 (Rajpurkar et al., 2016) is a popular benchmark dataset for evaluating the performance of question answering models. It was released by Stanford University in 2016 and contains over 100,000 questions with their corresponding answers, all based on a set of Wikipedia articles. After preprocessing, the dataset contains 98K <answer, passage, question> data triples. Input is formatted as <answer [SEP] passage> following GLGE. The average input and output lengths are 149.4 and 11.5, respectively.

4.2 Results

In this section, we show large scale pretraining using prefix-lm leads to huge improvement in performance for NAR decoder-only models. We evaluate our approach on two popular datasets. For XSum dataset, we use a combination of ROUGE score (Lin, 2004) to evaluate different models. As shown in table 1, we observe +2.9 ROUGE-L score when the model is pretrained. Also the model outperformed both BANG NAR and Semi-NAR and CMLM in terms of all three ROUGE metrics. We also evaluated our approach on Squad 1.1 question generation task, our model was able to show +14.4 ROUGE-L and +8.6 BLEU-4 when the model is pretrained. And it demonstrates +1.7 ROUGE-L improvement in performance compared to BANG NAR while -2.7 ROULGE-L compared to BANG semi-NAR.

5 Ablation Studies

5.1 Model Architecture

We conduct preliminary experiments on WMT14 using EN-DE on both encoder-decoder and decoder only model. The max BLEU number for encoder-decoder and decoder only model have negligible difference while the encoder-decoder model has a high variance during eval. Hence we utilize the decoder only architecture for the main study on other downstream benchmarks.

Model	@500k	@1M	best
Decoder only Pretrained	21.6	21.76	21.76
Encoder Decoder Pretrained	20.13	18.42	21.73

Table 2: BLEU on WMT14 EN→DE.

5.2 Sample Efficiency

Model	@500k	@1M
Decoder only From scratch	14.57	21.89
Decoder only Pretrained	21.6	21.76

Table 3: Decoder-only BLEU the WMT14 EN→DE.

In Table 3, we present the WMT14 ENDE numbers for pretrained vs from scratch numbers. We see although the final numbers have negligible difference, the pretrained model is more sample efficient, reaching higher number with the same fine-tune steps. Note that this number is not comparable with SoTA WMT results because of the length predictor, for fair comparison, please refer to SUNDAE Appendix Figure 4a.

6 Conclusion and Future Work

In this work, we investigate the effect of pretraining for non-autoregressive decoder only SUNDAE. We show that pretraining should be considered a foundational block for non-autoregressive model. For future work, there is a natural question: Will the non-autoregressive model scales with data size and model parameters as larger autoregressive models do.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. 2018. Jax: composable transformations of python+ numpy programs.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019a. Mask-predict: Parallel decoding of conditional masked language models.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019b. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu and Xiang Kong. 2020. Fully non-autoregressive neural machine translation: Tricks of the trade.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. *Advances in Neural Information Processing Systems*, 32.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020. Incorporating bert into parallel sequence decoding with adapters. *Advances in Neural Information Processing Systems*, 33:10843–10854.

Suzana Ilić, Edison Marrese-Taylor, Jorge A Balazs, and Yutaka Matsuo. 2018. Deep contextualized word representations for detecting sarcasm and irony. *arXiv preprint arXiv:1809.09795*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*.

Pengfei Li, Liangyou Li, Meng Zhang, Minghao Wu, and Qun Liu. 2022. Universal conditional masked language pre-training for neural machine translation. *arXiv preprint arXiv:2203.09210*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, et al. 2020. Glge: A new general language generation evaluation benchmark. *arXiv preprint arXiv:2011.11928*.

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. *arXiv preprint arXiv:1808.08745*.

OpenAI. 2023. Gpt-4 technical report.

Weizhen Qi, Yeyun Gong, Jian Jiao, Yu Yan, Weizhu Chen, Dayiheng Liu, Kewen Tang, Houqiang Li, Jiusheng Chen, Ruofei Zhang, et al. 2021. Bang:

Bridging autoregressive and non-autoregressive generation with large scale pretraining. In *International Conference on Machine Learning*, pages 8630–8639. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. 2022. Scaling up models and data with `t5x` and `seqio`.

Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. 2021. Step-unrolled denoising autoencoders for text generation. *arXiv preprint arXiv:2112.06749*.

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. *arXiv preprint arXiv:2102.08220*.