Language Understanding with Non-Autoregressive BERT-to-BERT Autoencoder

Mohammad Golam Sohrab[1] Matīss Rikters[1] Makoto Miwa[1,2]
[1]Artificial Intelligence Research Center (AIRC)
National Institute of Advanced Industrial Science and Technology
[2]Toyota Technological Institute, Japan
{sohrab.mohammad, matiss.rikters}@aist.go.jp
makoto-miwa@toyota-ti.ac.jp

Abstract

We propose a fast and scalable non-autoregressive S2S model that is flexible enough to employ the widely successful BERT as the backbone for the encoder and the decoder. In this paper, we pre-train BERT models using the non-autoregressive approach, which we call non-autoregressive BERT-to-BERT (NAR-B2B). We evaluate the NAR-B2B model using the standard GLUE tasks for natural language understanding. Experimental results show that the pre-training strategy is effective for BERT and the non-autoregressive approach enables fast training and decoding.

1 Introduction

Pre-trained language models (PLMs) have been widely successful across many natural language processing (NLP) tasks. Especially, the Bidirectional Encoder Representations from Transformers (BERT) model [1] has received widespread attention due to its ability to infer contextualised word representations. With this ability, BERT can help obtain superior performance in downstream tasks by leveraging simple fine-tuning. Recently, pre-training sequence-to-sequence (S2S) models such as BERT2BERT [2], BART [3], T5 [4], and Optimus [5] have been introduced. They are applied to various tasks in NLP since they are well suited to problems such as summarisation and question answering. These models employ transformer-based bidirectional models, such as BERT, as an encoder and use as a decoder powerful auto-regressive models like GPT-2 [6], which generate tokens in a left-to-right manner. Among these models, Optimus is based on variational autoencoders (VAE) [7, 8], which can be considered the first large-scale pre-trained deep latent variable model and proven useful for non-autoregressive modeling [9].

These models, however, have several limitations. First, the models need many computational resources to decode, especially for longer texts. For instance, Optimus limited their inputs to the length of 64, and it also limited the training data. Second, some models use different architectures for their encoders and decoders. For example, Optimus is based on BERT and GPT-2. Such inconsistency in models results in different tokenisation between input and output.

To tackle these limitations, we aim at building a large-scale non-autoregressive pre-trained S2S model using the BERT as the backbone for both encoder and decoder models. The non-autoregressive modelling allows fast decoding and the use of longer texts. Unlike the Optimus and BART models, our model does not rely on autoregressive GPT-2-like models. The input and output tokenisation become consistent by using the same BERT architecture.

In this paper, as the first step towards the above modelling, we propose to pre-train the BERT models using the non-autoregressive sequence-to-sequence model, which we call non-autoregressive BERT-to-BERT, or NAR-B2B. We train the entire non-autoregressive S2S model with an autoencoding objective, which is inspired by BERT. To investigate better autoencoding, we compare autoencoder (AE) and variational autoencoder (VAE) models in the S2S modelling following Optimus. We evaluate the performance of the pre-trained NAR-B2B models to confirm that the BERT models can be trained via S2S modelling. We also evaluate and compare the pre-trained NAR-B2B

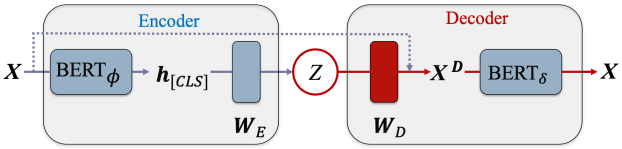

Figure 1 The NAR-B2B architecture (inspired by Optimus). The dotted line illustrates the ground truth words (GTW) setting where the decoder receives GTW as input.

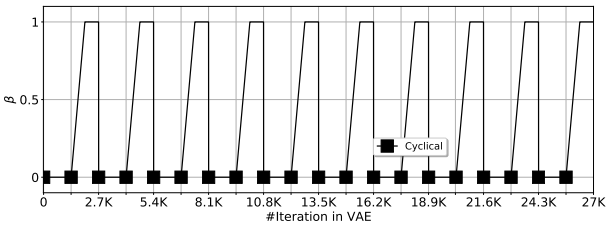

Figure 2 Beta (β) Annealing

models in the encoder and decoder. Experimental results show the effectiveness of our modelling in both speed and performance. First, we could pre-train the NAR-B2B models with more training data and less training time using the S2S model. Second, our pre-trained NAR-B2B models in both encoder and decoder perform better than the Optimus-encoder-based BERT and the original BERT model.

2 Pre-training NAR-B2B

To pre-train non-autoregressive BERT-to-BERT (NAR-B2B) models using the S2S modelling, we train the $\{\phi, \delta\}$ parameters, which corresponds to the parameters of the BERT encoder and the non-autoregressive (NAR) BERT decoder using an autoencoding objective, i.e., AE or VAE. The architecture of the proposed model is illustrated in Figure 1.

2.1 Model Architecture

The model architecture of NAR-B2B is composed of a multi-layer Transformer-based encoder and decoder, in which the embedding layer and the stack of transformer layers are initialised with BERT [1]. To leverage the expressiveness power of existing pre-trained BERT models, we initialise our BERT encoder BERT$_\phi$ and BERT decoder BERT$_\delta$ with the pre-trained BERT parameters. Here, we denote the number of layers (i.e., Transformer blocks) as L, the hidden size as H, and the number of self-attention heads as A. Specifically, in this paper, we employ bert-base-cased (L=12, H=768, A=12, Total Parameters=110M) for both encoder and decoder models.

Input Sequence Following the BERT setup, we first append a `[CLS]` and a `[SEP]` token on both sides of the source sentence. Then, inspired by the input format representation [10], we merge sentences for faster training and flexibility in dealing with longer sequences. Specifically, we merge multiple sentences with a special token `[SEP]` to ensure that the source sequence length is not longer than the BERT default maximum sequence length, which is 512.

BERT Encoder Our NAR-B2B model first feeds the input sequence $\boldsymbol{X} = \boldsymbol{x}_{[CLS]}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ to the BERT$_\phi$ embedding layer. In BERT, the first token of every sentence is a special classification token (`[CLS]`). The last layer's hidden state $\boldsymbol{h}_{[CLS]} \in \mathbb{R}^H$ corresponding to this token is used as the sentence-level representation. It further constructs the latent representation $\boldsymbol{z} = \boldsymbol{W}_E \boldsymbol{h}_{[CLS]}$ where $\boldsymbol{z} \in \mathbb{R}^P$ is a P-dimensional vector and $\boldsymbol{W}_E \in \mathbb{R}^{P \times H}$ is the weight matrix.

Building Decoder Input The latent embedding $\mathbf{z}$ is added to the `[PAD]` and positional embeddings to construct the decoder's input sequence $\boldsymbol{X}^D$. In addition to this PAD setting, we tried to use the input sequence of the encoder to construct the input as shown as the dotted line in Figure 1[1)]. Under this setting, the decoder receives the encoder's ground truth words (GTW) as input. The embedding representation $\boldsymbol{x}_i^D$ of the i-th token of the decoder's input sequence is calculated as $\boldsymbol{x}_i + \mathbf{W}_D \mathbf{z}$. To facilitate $\mathbf{z}$ in BERT decoding, we add the latent vector $\mathbf{z}$ to all inputs of the intermediate layers in the decoder.

Non-autoregressive BERT Decoding Our non-autoregressive BERT decoder BERT$_\delta$ receives the input $\boldsymbol{X}^D$ and generates the original sequence in a non-autoregressive manner (all tokens simultaneously).

2.2 Training

To train NAR-B2B, we employ AE and VAE objectives.

AE To train NAR-B2B with the AE objective, the model entirely focuses on maximizing the mutual information (MI) [5, 11] to recover a sentence from the latent space.

VAE The VAE objective consists of two terms, (a) reconstruction and (b) Kullback-Leibler (KL) regularization, balanced by a weighting hyper-parameter β. When training with the KL regularization, the KL tends to vanish [11, 5]. [11] proposed a cyclical annealing schedule (CAS), which

1) Since our target is the pre-training of BERT, we tried to use the encoder's input sequence in the decoder.

repeats the annealing process of beta multiple times by showing that the KL vanishing is caused by the lack of good latent codes in the training decoder at the beginning of optimization. To train NAR-B2B with the VAE objectives, we progressively improve latent representation $\mathbf{z}$ by adapting a CAS [11], where β is repeatedly annealed from 0 to 1. We split the training iterations into ten cycles, starting with $\beta = 0$ and ending with $\beta = 1$. Within each cycle, there are three consecutive stages: training AE ($\beta = 0$) for 0.5 proportion, annealing β from 0 to 1 for 0.25 proportion, and fixing $\beta = 1$ for 0.25 proportion (as illustrated in Figure 2).

3 Experiments

The pre-training procedure follows existing literature on PLM pre-training. Following BERT, we use BookCorpus[2] [12] and English Wikipedia[3] to pre-train our NAR-B2B model. (details in Appendix B).

Table 1 shows the performance comparison of our NAR-B2B model with BERT [1] and Optimus [5]. The NAR-B2B model yields higher performance than the BERT and Optimus models. BERT and Optimus (VAE) are considered baseline models, and their scores are taken from the Optimus [5] literature. In this table, the scores in the bold text indicate the best performance of a certain task in GLUE datasets. The results of the pre-trained encoder model of NAR-B2B using the AE objectives (NAR-B2B (AE) + Encoder) show that the model performs higher than the baseline models on five tasks. The pre-trained decoder model of NAR-B2B using the AE objectives (NAR-B2B (AE) + GTW + Decoder) outperforms the baseline models on all tasks except QQP. Since our NAR-B2B (AE) + GTW + Decoder models outperform the NAR-B2B (AE) + Encoder model, we evaluate the pre-trained decoder model of NAR-B2B with the VAE objectives (NAR-B2B (VAE) + Decoder model) and report the results in Table 1, where the model outperforms the baseline models.

Table 2 shows the performance of the NAR-B2B decoder using the AE, full-VAE, VAE with GTW, and VAE with PAD on a sample dataset (10% from 207M data). For the full-VAE model, we annealed β from 0.25 to 1 to let the model always learn only in VAE objectives. During β annealing, we follow the same proportion as the cyclical schedule in VAE stated in Section 2.2. This table shows the effectiveness of AE and VAE models with different settings over the baselines and full-VAE approaches, where VAE with PAD achieves the best score in terms of average in the four tasks of the GLUE tasks. Our NAR-B2B with AE and VAE achieve competitive scores as the original BERT, compared with the VAE with the PAD model. Our NAR-B2B model initialised with sample data perform comparable scores of Table 1.

Training large models on large-scale data sets is computationally challenging. In Optimus [5], computational efficiency is missing to train the model. Therefore, in our earlier attempt, we train the Optimus using 32M sentences that needs around 20K iterations using a sequence length of 64 for 46 hours. In contrast to training the NAR-B2B model, it needs around 27K iterations using a sequence length of 512 using 207M sentences for 39 hours. Our model can learn around seven times larger data than Optimus with lower cost while improving performance (details in Appendix A). Experimental results based on GLUE tasks with different objectives of NAR-B2B have demonstrated strong performance of NAR-B2B models. Besides, NAR-B2B can efficiently learn and decode the large-scale dataset by providing longer sequences. In future extensions, it would be interesting to investigate how effectively we can train our PLM in several hours without degrading the performance by adapting efficient transformers like Big Bird [13], where the maximum sequence length is 4096.

We report the performance of WNLI based on 634 examples of the GLUE dataset separately in Table 3 for readers' interest. The GLUE benchmark consists of nine datasets, but BERT [1] literature ignores the WNLI dataset because of its problematic nature, as they reported. The Optimus literature reports the Optimus performance that includes WNLI; meanwhile, Optimus reports the score along with the WNLI dataset by re-running the BERT model. The Optimus model performs lower than BERT but outperforms the BERT model while adding the WNLI score. In this table, our NAR-B2B model with ground truth words (GTW) as input to the decoder model outperforms the BERT and Optimus model [4]. In contrast, our NAR-B2B model with PAD setting as decoder input outperforms the BERT model

2) https://github.com/huggingface/datasets/tree/master/datasets/bookcorpus

3) https://github.com/huggingface/datasets/tree/master/datasets/wikipedia

4) In decoder input, the Optimus model follows the GTW setting, while PAD setting in Optimus is absent.

PLM	MNLI 392k ACC	QQP 363k ACC	QNLI 108k ACC	SST-2 67k ACC	COLA 8.5k MCC	STS-B 5.7k P.C.	MRPC 3.5k F1	RTE 2.5k ACC	Avg.
BERT	0.835	**0.909**	0.912	0.923	0.598	0.886	0.868	0.700	0.829
Optimus (VAE)	0.834	**0.909**	0.908	0.924	0.573	0.888	0.873	0.697	0.825
NAR-B2B (AE) + GTW + Encoder	0.839(+)	0.908(-)	0.911(-)	0.929(+)	0.608(+)	0.887(-)	**0.897**(+)	**0.711**(+)	0.836(+)
NAR-B2B (AE) + GTW + Decoder	0.838(+)	0.906(-)	0.913(+)	0.928(+)	**0.619**(+)	**0.897**(+)	0.893(+)	0.704(+)	**0.837**(+)
NAR-B2B (AE) + PAD + Decoder	0.830(-)	0.905(-)	0.911(-)	**0.931**(+)	0.591(-)	0.890(+)	0.891(+)	0.690(-)	0.830(+)
NAR-B2B (VAE) + GTW + Decoder	0.838(+)	0.908(-)	0.910(-)	0.928(+)	0.606(+)	**0.897**(+)	0.886(+)	0.686(-)	0.832(+)
NAR-B2B (VAE) + PAD + Decoder	**0.841**(+)	0.904(-)	**0.915(+)**	0.926(+)	0.608(+)	0.894(+)	0.889(+)	0.693(-)	0.834(+)

Table 1 Comparison of BERT, Optimus, and NAR-B2B with the AE and VAE objectives on the validation set of GLUE. (+) denotes our models performance being higher than baseline models, (-) indicates our models performance being lower. PAD denotes the decoder receives the BERT special token [PAD] as its input, while GTW denotes the decoder receives the ground truth words as its input.

PLM	MNLI 392k ACC	QQP 363k ACC	QNLI 108k ACC	SST-2 67k ACC	COLA 8.5k MCC	STS-B 5.7k P.C.	MRPC 3.5k F1	RTE 2.5k ACC	Avg.
NAR-B2B (AE) + GTW + Decoder	**0.841**	**0.907**	0.908	0.924	0.607	0.895	0.890	0.672	0.831
NAR-B2B (full-VAE) + GTW + Decoder	0.836	**0.907**	**0.912**	0.923	**0.616**	0.891	0.883	0.628	0.825
NAR-B2B (VAE) + GTW + Decoder	**0.841**	**0.907**	0.911	0.926	0.601	0.894	0.885	0.672	0.830
NAR-B2B (VAE) + PAD + Decoder	0.838	0.902	0.908	**0.935**	0.611	**0.896**	**0.893**	**0.711**	**0.837**

Table 2 Comparison of NAR-B2B model on a sample dataset (10% from 207M sequences). NAR-B2B performances with the AE, full-VAE, VAE, and VAE with PAD objectives. Comparison is on the validation set of GLUE.

PLM	Avg. Score Table 1	WNLI (ACC) 634
BERT	0.829	0.507
Optimus (VAE)	0.825	0.563
NAR-B2B (AE) + GTW	**0.837**	**0.620**
NAR-B2B (VAE) + PAD	0.834	0.563

Table 3 Comparison of BERT, Optimus (VAE), and NAR-B2B with the WNLI dataset.

but exact score to the Optimus model.

4 Related Work

Pre-trained Language Models (PLMs) are neural networks trained on large-scale datasets that can be fine-tuned on problem-specific data. Some of the most popular have been GPT-2 [6], XLNet [14], and XLM [15]. They became widely adapted after BERT [1] reported SOTA results for 11 NLP tasks.

Li et al. [5] proposed the first large-scale language VAE model, Optimus. They connect a BERT encoder and a GPT-2 decoder using a universal latent embedding space. The model is first pre-trained on a large text corpus and then fine-tuned for various language generation and understanding tasks. It achieves SOTA on VAE language modelling benchmarks.

Rothe et al. [2] developed a Transformer-based sequence-to-sequence models by describing several combinations of model initialization that include BERT2BERT, a BERT-initialized encoder paired with a BERT-initialized autoregressive decoder. Our implementation and architecture of NAR-B2B is a non-autoregressive VAE-based model without cross-attention which differs from BERT2BERT.

The Cyclical Annealing Schedule [11] was proposed to mitigate the problem of KL regularisation vanishing. They increase the β weighting hyper-parameter multiple times, which progressively enables learning of more meaningful latent codes by leveraging the results of previous learning cycles as a warm restart.

NAR models have been recently investigated in NLP tasks due to their efficiency. For example, Gu et al. [16] introduced a NAR model for machine translation (MT) based on the transformer [17]. This reduced latency by 90% and achieved competitive output quality with only a slight decrease in translation performance compared to similar-sized autoregressive (AR) models. Furthermore, Gu and Kong [9] further minimised the gap with AR models achieving SOTA performance on several MT benchmarks.

5 Conclusion

In this paper, we introduced a fast and scalable S2S model non-autoregressive BERT-to-BERT (NAR-B2B) that aims at building a large-scale NAR pre-trained S2S model using BERT as the backbone for both encoder and decoder models. To investigate better modelling, we compared AE and VAE models in the S2S modelling. We evaluated the NAR-B2B model's contribution over the nine language understanding tasks, and the results show that the NAR-B2B model with different settings consistently performs better in comparison to the original BERT and Optimus models. In future work, we plan to expand the experiments to more language combinations for MT and additional language generation tasks.

Acknowledgement

This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] Jacob Devlin, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of NAACL**, pp. 4171–4186, 2019.

[2] Sascha Rothe, et al. Leveraging pre-trained checkpoints for sequence generation tasks. **TACL**, Vol. 8, pp. 264–280, 2020.

[3] Mike Lewis, et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **Proceedings of ACL**, pp. 7871–7880. ACL, 2020.

[4] Colin Raffel, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. **JMLR**, Vol. 21, pp. 1–67, 2020.

[5] Chunyuan Li, et al. Optimus: Organizing sentences via pre-trained modeling of a latent space. In **Proceedings of EMNLP**, pp. 4678–4699. ACL, 2020.

[6] Alec Radford, et al. Language models are unsupervised multitask learners. **OpenAI blog**, Vol. 1, No. 8, p. 9, 2019.

[7] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In **ICLR**, 2014.

[8] Danilo Jimenez Rezende, et al. Stochastic backpropagation and approximate inference in deep generative models. In **ICML**, pp. 1278–1286, 2014.

[9] Jiatao Gu and Xiang Kong. Fully non-autoregressive neural machine translation: Tricks of the trade. In **Findings of ACL-IJCNLP 2021**, pp. 120–133. ACL, 2021.

[10] Yinhan Liu, et al. Roberta: A robustly optimized bert pretraining approach. **arXiv preprint**, 2019.

[11] Hao Fu, et al. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In **Proceedings of NAACL**, pp. 240–250. ACL, 2019.

[12] Yukun Zhu, et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In **ICCV**, 2015.

[13] Manzil Zaheer, et al. Big bird: Transformers for longer sequences. **NeurIPS**, Vol. 33, , 2020.

[14] Zhilin Yang, et al. Xlnet: Generalized autoregressive pretraining for language understanding. In **NeurIPS**, Vol. 32, pp. 5753–5763, 2019.

[15] Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In **NeurIPS**, Vol. 32, pp. 7059–7069, 2019.

[16] Jiatao Gu, et al. Non-autoregressive neural machine translation. In **ICLR**, 2018.

[17] Ashish Vaswani, et al. Attention is all you need. **NIPS**, Vol. 30, pp. 6000–6010, 2017.

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **ICLR**, 2019.

[19] Alex Wang, et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In **Proceedings of the BlackboxNLP Workshop**, pp. 353–355. ACL, 2018.

A Computational Efficiency

Table 4 shows the computational efficiency of NAR-B2B in comparison to the Optimus model. We train the Optimus and NAR-B2B based on a single epoch. Optimus can use a larger batch size because of the shorter sequence length.

Model	Data	Batch	Iter.	GPUs	Time
Optimus	32M	1,600	20K	200	46H
NAR-B2B + AE	207M	600	27K	200	38.5H
NAR-B2B + VAE	207M	600	27K	200	39H

Table 4 Computational details

B Experimental Settings

B.1 Pre-training

The BookCorpus data set is already processed into 86M sentences. We load Wikipedia with the version of 20200501.*en* from huggingface datasets and split the text from `text` field into sentences by detecting newlines that lead to 121M sentences. We ignore the `title` field in Wikipedia. To pre-train our NAR-B2B model, our model receives 207M sentences from combining BookCorpus and Wikipedia datasets. Using sentence merging described in Section 2.1, it further compresses the data from 207M into 16M longer sequences split into 100 files as the input to the BERT model.

The NAR-B2B model we trained had 12 layers in the encoder and decoder. We used a batch size of 600 and trained the model for 13,302 steps. We optimized our PLM models using AdamW [18] with the learning rate of 5e-5 for BERT-base-cased. We trained each language model with 200 GPUs (NVIDIA V100 for NVLink 16GiB) with a batch size of 600. The maximum sequence length is set to 512.

B.2 Language Understanding

We consider the GLUE benchmark [19], which consists of nine datasets for general language understanding. Following the fine-tuning setting in [1, 5], we use the learning rate $[2, 3, 4, 5] \times 10^{-5}$ with different seeds and train the model for three epochs. We select the best performances among different runs. We employ Optimus evaluation script [5)] for GLUE to report all the scores.

5) https://github.com/ChunyuanLI/Optimus