Non-autoregressive Pre-trained Sequence-to-Sequence Modeling with BERT-NAR-BERT

Mohammad Golam Sohrab[1] Masaki Asada[1] Matīss Rikters[1] Makoto Miwa[1,2]
[1]National Institute of Advanced Industrial Science and Technology
[2]Toyota Technological Institute, Japan
{firstname.lastname}@aist.go.jp makoto-miwa@toyota-ti.ac.jp

Abstract

We introduce BERT-NAR-BERT (BnB) – a pre-trained non-autoregressive sequence-to-sequence model, which employs BERT as the backbone for the encoder and decoder for natural language generation tasks in general and biomedical domains. We adopt length classification and connectionist temporal classification models to control the output length of BnB. Evaluation results on language understanding, abstractive summarization, question generation, machine translation and biomedical text summarization show substantial improvements in inference speed (~10x) with a slight deficiency in output quality compared to our autoregressive baseline. Code is released on GitHub[1)] under the Apache 2.0 License.

1 Introduction

Sequence-to-sequence (S2S) models have recently been widely used for natural language processing problems. The S2S architecture is first introduced in the field of machine translation (MT) [1] and later used for pre-trained generative language models (LM), such as BART [2], Optimus [3] and BERT2BERT [4]. These models usually adopt an autoregressive (AR) decoding strategy to generate texts from left to right, token by token. AR decoding can perform high-quality inference, but it has the limitation that it cannot decode tokens in parallel and requires more time and computational cost for inference.

In this paper, we propose a novel S2S non-autoregressive (NAR) model based on existing Transformer [5] architectures to allow parameter initialization from publicly available PLM checkpoints. Specifically, we extend the BERT2BERT (B2B) [4] model to build a NAR S2S model using BERT as the backbone for both encoder and decoder models. The NAR modeling allows fast decoding and generation of longer texts. Using BERT as the backbone, we can start training with reliable parameters by loading the pre-trained BERT checkpoints. In addition, unlike the B2B model, we perform one epoch of additional pre-training starting at the BERT checkpoints and investigate the effectiveness of the additional pre-training[2)]. We adopt the Length Classification (LC) [6] or Connectionist Temporal Classification (CTC) [7] models to control the output length of BnB.

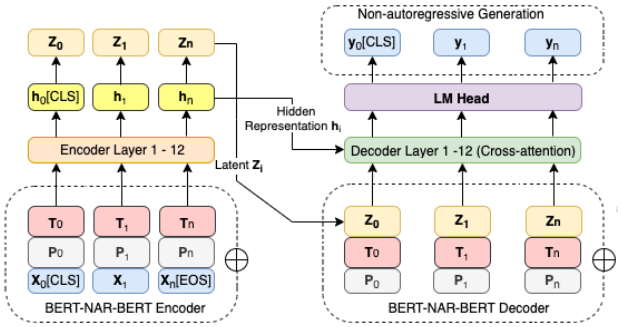

Figure 1 The S2S BERT-NAR-BERT (BnB) architecture.

We fine-tune and evaluate the BnB model on the general language understanding evaluation (GLUE), abstractive summarization, question generation, MT, and biomedical text summarization. We compare its performance with several AR and NAR models.

2 BERT-NAR-BERT Framework

Our model extends the B2B model into a NAR decoder with additional pre-training and modeling output length. It is our direct AR baseline. The overview of BnB is shown in Figure 1.

1) https://github.com/aistairc/BERT-NAR-BERT

2) We use the term additional pre-training for pre-training our BnB model. We load the pre-trained checkpoints and do not pre-train from scratch.

2.1 Model Architecture

The BnB model comprises a multi-layer Transformer-based encoder and decoder, in which the embedding layer and the stack of transformer layers are initialized with BERT [8]. To leverage the expressiveness power of existing pre-trained BERT models, we initialize our encoder and decoder parts with the pre-trained BERT parameters. 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.

BnB Encoder The encoder part of BnB is the same architecture as the BERT model. The BnB model first feeds the source input sequence $\boldsymbol{X} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ to the BnB encoder layer. In the BnB embedding layers, the input representation is constructed by summing the corresponding token (X), position (P), and type (T) embeddings.

The embeddings are fed into the BERT self-attention and feed-forward layers. The hidden representation of the final layer, $\boldsymbol{h}$, is passed to the subsequent layer for obtaining latent representations.

Latent Representations We construct the latent representation $z = \boldsymbol{W}_E \boldsymbol{h} + \boldsymbol{b}$ based on token-level representation from the encoder hidden state $\boldsymbol{h}$ where $z \in \mathbb{R}^P$ is a P-dimensional vector and $\boldsymbol{W}_E \in \mathbb{R}^{P \times H}$ is the weight matrix. A visualization of BnB encoder and the latent representations can be seen in the left part of Figure 1.

BnB Decoder The decoder part is also based on the BERT architecture, and we can directly initialize the decoder with the pre-trained BERT model. The cross-attention mechanism is adopted from BERT2BERT and the encoder hidden representation of the final layer $\boldsymbol{h}$ is used for cross-attention. Our model differs from BERT2BERT in input representation and attention masks to enable NAR decoding. In our BnB decoder, input representation is constructed without providing any target tokens. The input representation is constructed by summing the corresponding P and T embeddings and the latent embedding $\mathbf{z}$ from the encoder. The attention masks are the normal masks that give access to all future tokens. The resulting decoder output representations of the final layer are fed to the subsequent generation layer.

Masked Language Modeling We employ the self-supervised learning strategy adopted in BERT. We randomly mask the tokens in each sequence, and all masked tokens are predicted in a non-autoregressive manner.

Permutation Language Modeling We randomly permute tokens in a sequence and predict the original order of the tokens, which is inspired by the idea in XLNet [9].

2.2 Modeling Output Length

To control the generation output length of BnB, we implement two length prediction (LP) models, namely: 1) Length Classification [6], and 2) CTC [7] that implicitly determines the target length from the token alignment.

Length Classification The length classification formulates the LP as a classification task and utilizes the latent representations to predict the target length:

$$p_\theta(\boldsymbol{y}|\boldsymbol{z}) = \sum_l p_\theta(\boldsymbol{y}, l|\boldsymbol{z}) = p_\theta(\boldsymbol{y}, l_y|\boldsymbol{z}) = p_\theta(\boldsymbol{y}|\boldsymbol{z}, l_y)\, p_\theta(l_y|\boldsymbol{z}), \quad (1)$$

where l_y denotes the length of $\boldsymbol{y}$ that is the gold length in training and $p_\theta\left(l_y|\boldsymbol{z}\right) = \mathcal{L}_L$ is the length predictor that predicts the length of target sentence $\boldsymbol{y}$. Once the sentence length is predicted, the model predicts each output token with a token-level categorical cross-entropy loss.

Connectionist Temporal Classification In contrast to LC, we also adopt the CTC [7] considering its superior performance and flexibility for latent alignment. The CTC loss is independently computed after the decoder by replacing the length classification and CE loss.

3 Experiments

In this section, we evaluate the BERT-NAR-BERT both in the general and biomedical domains over the different downstream tasks including GLUE, abstractive summarization, question generation, MT, and biomedical text summarization. (details in Appendix A).

3.1 GLUE

In Table 1, we compare our fully non-autoregressive BnB approach with the OpenAI GPT [3] [10], encoder-based BERT, and variational auto encoder-based Optimus models. In this table, we compare the BnB based on the permutation LM objectives. Our best BnB model with permutation LM objective function outperforms the OpenAI GPT [10], and BERT models on averaged scores, but shows comparable results with of Optimus. The Optimus model follows the sentence-level pre-training procedure which allows training the model longer than document-level.

3) The first version since the model parameters are comparable with the proposed BnB model.

Model	MNLI 392k ACC	QQP 363k F1/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.
OpenAI GPT [10]	82.1	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT [8]	**84.6**	71.2	90.5	**93.5**	52.1	85.8	**88.9**	66.4	79.6
Optimus [3]	83.4	**90.9**	**90.8**	92.4	**57.3**	**88.8**	87.3	**69.7**	**82.5**
BnB Encoder + additional pre-training	82.2	90.3	89.8	90.7	54.3	87.9	88.0	66.8	81.3

Table 1 Comparison of OpenAI GPT, BERT, Optimus, and BnB on the validation set of GLUE. ACC and P.C stand for accuracy and Pearson correlation coefficient, respectively. The OpenAI GPT scores are reported by Devlin et al. [8]. Bold and underlined scores denote the best and second-best results.

Type	Model	XSum				SQuAD v1.1		
		R-1	R-2	R-L	Latency (↓)	R-L	B-4	Latency (↓)
AR	Transformer [5]	30.7	10.8	24.5	18.33x	29.4	4.6	11.17x
	BART [2]	38.8	16.2	30.6	12.92x	42.6	17.1	8.00x
	BERT2BERT [4]	37.5	15.2	30.1	10.67x	39.3	13.5	10.67x
Semi-AR	iNAT [11]	27.0	6.9	22.4	2.17x	32.3	3.2	2.25x
	CMLM [12]	29.1	7.7	23.0	7.92x	29.6	3.9	7.50x
NAR	NAT [13]	24.0	3.9	20.3	1.25x	31.5	2.5	1.17x
	CMLM [12]	23.8	3.6	20.2	1.17x	31.6	2.5	1.17x
	ELMER-Hard [14]	34.5	9.8	26.1	0.83x	37.9	11.8	0.83x
	ELMER-Soft [14]	**38.3**	**14.2**	29.9	0.83x	40.2	13.5	0.83x
	BnB	32.7	11.6	27.8	1.0x	36.8	9.1	1.0x
	BnB + additional pre-training	36.1	13.4	**30.0**	1.0x	**41.7**	**13.8**	1.0x

Table 2 Performance comparison on XSum and SQuAD v1.1. R-1/2/L and B-4 stand for ROUGE-1/2/L and BLEU-4, respectively. Bold and underlined scores denote the best and second-best results within NAR models. We include existing method latency values reported by Li et al. [14].

3.2 Summarization and Question Generation

Table 2 shows the performance comparison of BnB over the XSum and SQuAD v1.1 datasets for summarization and question generation tasks, respectively. Our best model outperforms all the semi-autoregressive (semi-AR) and most of the NAR approaches but closely competes with the AR approaches. Under the NAR setting, we achieve second-best results over the XSum and SQuAD v1.1 datasets. Our model outperforms the ELMER-Hard on the XSum and SQuAD v1.1 datasets, while ELMER-Soft remains the definitive state-of-the-art. ELMER-Hard and ELMER-soft denote fine-tuning ELMER with hard and soft early exit strategies, respectively. Unlike BnB that needs to determine length by integrating an extra LP model, ELMER dynamically adjusts the output length by emitting an end token at any position with early exit strategies.

3.3 Machine Translation

For MT, we compare the scores with autoregressive baselines - Transformer MT models and BERT2BERT, initialized with random parameters or pre-trained multilingual BERT. We compare random initialization of parameters with initialization from mBERT, as well as knowledge distillation [15], which has previously proven to be highly beneficial for NAR MT models [16]. Results of MT experiments are summarized in Table 3. The results show that our model can compete with the baseline Transformer and B2B models after knowledge distillation. It is expected as the BnBs are compared with the autoregressive baseline translation models.

Model	EN - DE →	EN - DE ←	EN - RO →	EN - RO ←
Transformer	27.30	25.36	21.53	27.81
B2B mBERT	25.80	23.40	**23.24**	**30.67**
BnB random	7.15	8.02	4.12	7.11
BnB mBERT	6.81	11.07	5.92	9.36
BnB mBERT dist.	**27.49**	**27.06**	18.94	30.42

Table 3 MT experiment results in BLEU scores of training Transformer, B2B, and BnB initialized with multilingual BERT (mBERT); trained on original and distilled (dist.) data from WMT 2014 (German) and WMT 2016 (Romanian). Bold and underlined scores denote the best and second-best results.

3.4 Biomedical Text Summarization

Table 4 shows the performance comparison between AR methods and our BnB. First, our model is about 18 times faster than AR models in inference. Regarding genera-

Model	iCliniq			
	R-1	R-2	R-L	Latency
BART base [2]	61.43	48.68	59.71	18.5x
BioBART base [17]	61.07	48.47	59.42	18.7x
BnB	57.43	43.99	55.24	1.0x
	HealthCareMagic			
	R-1	R-2	R-L	Latency
BART base [2]	46.81	26.19	44.34	18.1x
BioBART base [17]	46.67	26.03	44.11	18.9x
BnB	40.05	19.38	38.27	1.0x

Table 4 The main results on the summarization tasks in the biomedical domain. R-1/2/L stand for ROUGE-1/2/L.

tive performance, biomedical BnB showed the ROUGE-1 scores of about 94% and 85% of the performance by BioBART on the iCliniq and HealthCareMagic datasets, respectively.

3.5 Inference Speedup

We also compare the differences in inference speed between the models. While training speed is mostly similar for all, BnB can generate output 17x faster on average due to its non-autoregressive nature (tested on NVIDIA V100 and A100 GPUs) as shown in the Latency column of Table 2. For MT, translating 2,000 sentences in the WMT16 test data set with BnB takes 87 seconds on a GPU and 587 seconds on a CPU. The same took 234 seconds on a GPU and 1,234 seconds on a CPU for an equivalent with the direct B2B baseline model. BERT-NAR-BERT consists of 110M parameters for the encoder and decoder, including 12 layers, 768 hidden sizes, and 12 self-attention heads. Our direct baseline BERT2BERT model follows the same parameters, 220M totals for both the encoder and decoder models.

4 Related Work

Pre-trained Language Models such as GPT-2 [18], XL-Net [9], and XLM [19] are neural networks trained on large-scale datasets that can be fine-tuned on problem-specific data. They became widely adopted after BERT [8], which reported state of the art (SOTA) results for 11 NLP tasks. Our PLM BnB is also trained on a large-scale dataset and further fine-tuned on 12 specific NLP tasks, 10 of which overlap with BERT.

Li et al. [3] proposed the first large-scale variational auto encoder (VAE) language 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 modeling benchmarks. While the general idea of our work is similar, there are several core differences from this paper. Our model does not have a VAE and instead of the GPT-2 decoder we use the same BERT as in the encoder.

Rothe et al. [4] developed Transformer-based S2S models by describing several combinations of model initialization that include BERT2BERT, a BERT-initialized encoder paired with a BERT-initialized AR decoder. Our implementation of BnB is similar, except for the main differences of having a length prediction model, a latent representation from the encoder output layer, and a NAR decoder. The NAR decoder can decode tokens in parallel which drastically reduce the inference computational cost.

Sohrab et al. [20] describe BERT-NAR-BERT (BnB) – a pre-trained non-autoregressive sequence-to-sequence model, which employs BERT as the backbone for the encoder and decoder for natural language understanding and generation tasks. BnB shows substantial improvements in inference speed over the downstream tasks. This work is a paper summarizing on the basis of the original research methods and findings of BERT-NAR-BERT [20].

5 Conclusion

This paper introduces an efficient non-autoregressive S2S model BERT-NAR-BERT that outperforms baselines in most of the language understanding, summarization and question generation tasks. Still, it remains competitive in the quality of outputs when evaluated on MT tasks. However, our model with distilled data shows improvement over the baseline approaches. We find that using pre-trained BERT models as the encoder and decoder, along with CTC for LP and knowledge distillation for MT, helps improve the performance of language generation tasks.

In future work, we plan to experiment with replacing the BERT models with other pre-trained language models which can be used as encoders/decoders, as well as running broader evaluations on other S2S NLP tasks. We will also address large language models and decoder only models in our future direction.

Acknowledgment

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

References

[1] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In **EMNLP 2014)**, pp. 1724–1734, Doha, Qatar, October 2014. ACL.

[2] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **ACL 2020**, pp. 7871–7880, Online, July 2020. ACL.

[3] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. In **EMNLP 2020**, pp. 4678–4699, Online, November 2020. ACL.

[4] Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. **Transactions of the Association for Computational Linguistics**, Vol. 8, pp. 264–280, 2020.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. **Advances in neural information processing systems**, Vol. 30, , 2017.

[6] Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 34, pp. 8846–8853, 04 2020.

[7] Jindřich Libovický and Jindřich Helcl. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In **EMNLP 2018**, pp. 3016–3021, Brussels, Belgium, October-November 2018. ACL.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **NAACL-HLT 2019**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. ACL.

[9] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In **Advances in Neural Information Processing Systems**, Vol. 32, pp. 5753–5763, 2019.

[10] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. **OpenAI Blog**, 2018.

[11] Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In **EMNLP 2018**, pp. 1173–1182, Brussels, Belgium, October-November 2018. ACL.

[12] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In **EMNLP-IJCNLP 2019**, pp. 6112–6121, Hong Kong, China, November 2019. ACL.

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

[14] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. ELMER: A non-autoregressive pre-trained language model for efficient and effective text generation. In **EMNLP 2022**, pp. 1044–1058, Abu Dhabi, United Arab Emirates, December 2022. ACL.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. **NIPS 2014 Deep Learning Workshop**, 2014.

[16] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In **EMNLP 2016**, pp. 1317–1327, Austin, Texas, November 2016. ACL.

[17] Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. BioBART: Pretraining and evaluation of a biomedical generative language model. In **Proceedings of the 21st Workshop on Biomedical Language Processing**, pp. 97–109, Dublin, Ireland, May 2022. ACL.

[18] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf, 2019. Accessed: 2023-08-19.

[19] Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In **Advances in neural information processing systems**, Vol. 32, pp. 7059–7069, 2019.

[20] Mohammad Golam Sohrab, Masaki Asada, Matīss Rikters, and Makoto Miwa. Bert-nar-bert: A non-autoregressive pre-trained sequence-to-sequence model leveraging bert checkpoints. **IEEE Access**, Vol. 12, pp. 23–33, 2024.

[21] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. **Bioinformatics**, Vol. 36, No. 4, pp. 1234–1240, 09 2019.

[22] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In **Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP**, pp. 353–355, Brussels, Belgium, November 2018. ACL.

[23] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In **EMNLP 2018**, pp. 1797–1807, Brussels, Belgium, October-November 2018. ACL.

[24] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In **Text Summarization Branches Out**, pp. 74–81, Barcelona, Spain, July 2004. ACL.

[25] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In **EMNLP 2016**, pp. 2383–2392, Austin, Texas, November 2016. ACL.

[26] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In **ACL 2017**, pp. 1342–1352, Vancouver, Canada, July 2017. ACL.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In **ACL 2002**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. ACL.

[28] Matt Post. A call for clarity in reporting BLEU scores. In **Proceedings of the Third Conference on Machine Translation: Research Papers**, pp. 186–191, Brussels, Belgium, October 2018. ACL.

[29] Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, Hongchao Fang, Penghui Zhu, Shu Chen, and Pengtao Xie. MedDialog: Large-scale medical dialogue datasets. In **EMNLP 2020**, pp. 9241–9250, Online, November 2020. ACL.

A Experimental Settings

A.1 Pre-training: Data and Task Settings

We pre-train our BERT-NAR-BERT (additional pre-training) by loading the PLM checkpoints as S2S task. The additional pre-training procedure follows existing literature on PLM pre-training. The entire Wikipedia data dump with the version of `20220301.en` from Huggingface datasets[4] is used by only considering the `text` field for pre-training without any data filtering scheme. As a result, 6,458,670 input texts that include multiple sequences are truncated to a maximum sequence length of 512, and 3.2B tokens are used for our additional pre-training. We initialize BnB with `bert-base-cased` and update all the parameters of BnB's encoder and decoder, while the checkpoints are saved from both encoder and decoder outputs. We set the 15% probability of masking for masked LM and 50% probability of permuting for permutation LM.

Biomedical Pre-training As a parameter initialization, we loaded BioBERT [21] v1.1 base-cased checkpoints for the initial values of the encoder and decoder parts of BnB. We then performed the additional pre-training on the PubMed/MEDLINE abstract corpus[5]. This corpus contains 5.4B tokens of research article abstracts from the 2021 version of PubMed/MEDLINE. We used the same vocabulary as BioBERT [21] to tokenize the texts. We truncated all the input texts to 512 maximum sequence lengths following BioBART [17], which is an autoregressive biomedical pre-trained model. CTC loss and permutation language modeling are adopted for additional pre-training.

A.2 Fine-tuning: Data and Task Settings

For fine-tuning, we describe the data and task settings of the benchmark downstream tasks. We initialize all the downstream tasks with additional pre-training checkpoints generated from BnB.

GLUE We consider the General Language Understanding Evaluation (GLUE) benchmark [22], where we employ the Optimus evaluation script[6] to evaluate the scores and select the best performances among different runs to report all the scores following Optimus.

Abstractive Summarization Abstractive text summarization aims to produce a short version of a document while preserving its salient information content. We evaluate the models based on the BBC extreme [23] (XSum) dataset. This is a news summarization dataset containing 227K news articles and single-sentence summary pairs. We set the number of training epochs to 100 and adopt early stopping. The evaluation metric is ROUGE [24], including ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L). For latency comparison, we evaluate the time required to generate all the samples in the validation set with the same machine settings for our BnB and ELMER [14] and calculate the ratios of latency. We include the reported latency values of other existing models from ELMER.

Question Generation SQuAD v1.1 [25] is a dataset created for machine reading comprehension. The dataset contains 98K triples of {passage, question, answer} where the input is formatted as answer [SEP] passage. We use this data as a question generation dataset, in which a model receives an answer and a passage and generates the corresponding question. We follow the same train, validation, and test data split setting as Du et al. [26]. The evaluation metrics are ROUGE-L and BLEU-4 (B-4). We follow the same settings as the abstractive summarization task for latency comparison.

Machine Translation We evaluate our models using data sets from the WMT shared tasks on news translation - English (EN) ↔German (DE) data from WMT 2014 and English↔Romanian (RO) data from WMT 2016. We also experiment with distilled versions of these data sets generated by vanilla transformer models [5] trained on the normal data. We load the WMT datasets from Huggingface datasets[7),8)] and use them directly to train the models without filtering, back-translation, or any other kinds of synthetic data generation. We evaluate the performance by computing BLEU [27] scores using sacreBLEU [28].

Biomedical Text Summarization For fine-tuning the model on downstream tasks, we follow the same settings as BioBART [17]. We fine-tuned our model on two biomedical text summarization tasks in English: iCliniq and HealthCareMagic [29] datasets.

4) https://huggingface.co/datasets/wikipedia
5) https://huggingface.co/datasets/pubmed
6) https://github.com/ChunyuanLI/Optimus
7) https://huggingface.co/datasets/wmt14
8) https://huggingface.co/datasets/wmt16