Binary and Ternary Natural Language Generation

Zechun Liu*
Reality Labs, Meta Inc.
zechunliu@meta.com

Barlas Oğuz*
Meta AI
barlaso@meta.com

Aasish Pappu
Meta AI
aasish@fb.com

Yangyang Shi
Reality Labs, Meta Inc.
yyshi@meta.com

Raghuraman Krishnamoorthi
Reality Labs, Meta Inc.
raghuraman@meta.com

Abstract

Ternary and binary neural networks enable multiplication-free computation and promise multiple orders of magnitude efficiency gains over full-precision networks if implemented on specialized hardware. However, since both the parameter and the output space are highly discretized, such networks have proven very difficult to optimize. The difficulties are compounded for the class of transformer text generation models due to the sensitivity of the attention operation to quantization and the noise-compounding effects of autoregressive decoding in the high-cardinality output space. We approach the problem with a mix of statistics-based quantization for the weights and elastic quantization of the activations and demonstrate the first ternary and binary transformer models on the downstream tasks of summarization and machine translation. Our ternary BART base achieves an R1 score of 41 on the CNN/DailyMail benchmark, which is merely 3.9 points behind the full model while being 16x more efficient. Our binary model, while less accurate, achieves a highly non-trivial score of 35.6. For machine translation, we achieved BLEU scores of 21.7 and 17.6 on the WMT16 En-Ro benchmark, compared with a full precision mBART model score of 26.8. We also compare our approach in the 8-bit activation setting, where our ternary and even binary weight models can match or outperform the best existing 8-bit weight models in the literature. Our code and models are available at: https://github.com/facebookresearch/Ternary_Binary_Transformer.

1 Introduction

Generative pre-trained transformers (Brown et al., 2020; Lewis et al., 2020; Radford et al., 2018) have emerged as powerful and generic tools, driving breakthroughs not only in language understanding but the field of AI in general. These models owe their success mainly to their seemingly infinite ability to scale to ever-larger data and model sizes. Unfortunately, such scaling comes at the cost of large computational requirements, putting extensively large generative transformers out of reach of all but the most resource-rich institutions. Even moderately sized pre-trained transformers have limited applications due to their size and computational cost. Making generative transformers more efficient is imperative for widening their use to more devices and practical applications.

In this work, we explore making generative pre-trained transformers more efficient via the quantization of their weights and activations. Quantizing the weights of a neural network is useful for compression and allows the model to be stored more efficiently. However, compression alone does not reduce computation costs since the network's activations need to be computed in full precision. Quantizing both weights and activations allows computation to be performed with lower precision, potentially leading to significant efficiency gains depending on the quantization level and hardware implementation. Quantizing neural networks have a long history, and multiple works have attempted to quantize pre-trained transformers at various quantization levels (Shen et al., 2020; Zhang et al., 2020; Liu et al., 2022; Qin et al., 2021). Most of this work focuses on encoder-only models (mainly BERT) for sentence and token classification tasks. Quantizing text generation models has generally been regarded as a more difficult task (Behnke et al., 2021; Tao et al., 2022) due to the large output vocabulary and sequential decoding. Recent work has tackled this problem, though only for mild quantization levels (down to 8-bit activations) and with mixed success.

In contrast, we are interested in very low-bit quantization, down to ternary and even binary weights and activations. In order to achieve this, we combine and unify best practices for weight and activation quantization and present a frame-

*Equal contribution

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 65–77

work that uses gradient-matching quantization for weights and elastic quantization for activations. We apply our method to natural language generation tasks and, for the first time, demonstrate low-bit generative transformers of competitive accuracy. Our ternary (weight and activation) model lags a full-precision BART (Lewis et al., 2020) model by only 4 points in ROUGE on the XSUM summarization dataset. In contrast, our model with ternary weights and 8-bit activations comes within 1 point and even outperforms comparable state-of-the-art models with 8-bit weights. We also demonstrate a fully binary (weights and activations) model. While not as competitive, it is able to achieve a highly non-trivial ROUGE-1 score of 31.7.

Our results also extend to machine translation models. On the WMT16 En-Ro benchmark, we quantize an mBART model to extend the ternary-weight 8-bit activation SoTA by 1.2 points while demonstrating fully ternary and fully binary translation models for the first time.

We summarize our contributions as follows:

• We propose a novel combination of statistics-based weight quantization with learning-based activation quantization, which enables stably training transformer encoder-decoder models to converge in the fully ternary/binary settings, which was not previously possible.

• We significantly improve the state-of-the-art text generation models in the 8-bit activation and ternary/binary weight settings while setting the first non-trivial baselines for the fully ternary and fully binary settings.

2 Method

In this section, we first introduce the previous practices in binarization and ternarization. Then, we introduce a unified statistic-based weight binarization / ternarization method that can alleviate the gradient mismatch issue and enhance the quantized weights entropy. Lastly, we analyze the difference between weight quantization and activation quantization and propose an elastic ternarization method for activations. We abbreviate our method as TBT, short for "**T**ernary / **B**inary **T**ransformer".

2.1 Preliminary

2.1.1 Ternarization

Ternary neural networks, where real values are quantized to three levels, are first introduced in (Li et al., 2016). Thus, these values can be represented in 2 bits, leading to a 16× reduction in size and computation. Moreover, the computations can be calculated multiplication-free, leading to even further computation gains on suitable hardware. The recent work integrates the ternarization algorithm in natural language models for quantizing the weights and activations in classification tasks (Zhang et al., 2020) and ternarizing the weight (8-bit activations are used) in generative models (Li et al., 2022; Tao et al., 2022). The general formula (Li et al., 2016) for ternarization is as follows:

$$\mathbf{X}^i_{\mathbf{T}} = \begin{cases} -\alpha_{\mathbf{T}}, & \text{if } \mathbf{X}^i_{\mathbf{R}} < -\Delta \\ 0, & \text{if } -\Delta \leqslant \mathbf{X}^i_{\mathbf{R}} \leqslant \Delta \\ +\alpha_{\mathbf{T}}, & \text{if } \mathbf{X}^i_{\mathbf{R}} > \Delta \end{cases} \tag{1}$$

$$\Delta = \frac{0.7 \cdot ||\mathbf{X_R}||_{l1}}{n_{\mathbf{X_R}}} \tag{2}$$

$$\alpha_{\mathbf{T}} = \frac{\sum_i \mathbf{X}^i_{\mathbf{R}} \cdot \mathbf{1}_{|\mathbf{X}^i_{\mathbf{R}}|>\Delta}}{\sum_i \mathbf{1}_{|\mathbf{X}^i_{\mathbf{R}}|>\Delta}} \tag{3}$$

Here $\mathbf{X_T}$ denotes the ternary weights/activations, and $\mathbf{X_R}$ represents their real-valued counterparts. $n_{\mathbf{X_R}}$ denotes the total number of elements in the tensor. Δ is the ternary threshold, and $\alpha_{\mathbf{T}}$ is the scaling factor that minimizes l2-loss between $\mathbf{X_T}$ and $\mathbf{X_R}$.

2.1.2 Binarization

The neural network binarization denotes representing the weights and/or activation with bi-level values. It is first proposed in BNN (Courbariaux et al., 2016) and has evolved in the follow-up works (Rastegari et al., 2016; Liu et al., 2018). Rastegari et al. (2016) formulates binarization as:

$$\mathbf{X}^i_{\mathbf{B}} = \alpha_{\mathbf{B}} \cdot \text{Sign}(\mathbf{X}^i_{\mathbf{R}}) = \begin{cases} -\alpha_{\mathbf{B}}, & \text{if } \mathbf{X}^i_{\mathbf{R}} < 0 \\ +\alpha_{\mathbf{B}}, & \text{if } \mathbf{X}^i_{\mathbf{R}} \geqslant 0 \end{cases} \tag{4}$$

$$\alpha_{\mathbf{B}} = \frac{||\mathbf{X_R}||_{l1}}{n_{\mathbf{X_R}}} \tag{5}$$

Here $\mathbf{X_B}$ can represent binary weights or binary activations. $\alpha_{\mathbf{B}}$ denotes the scaling-factor that minimize the l2 loss between $\mathbf{X_R}$ and $\alpha_{\mathbf{B}} \cdot \text{Sign}(\mathbf{X_R})$.

The acceleration and compression effect of ternary/binary neural networks is significant. By representing the weights and activations with $\{-1, 0, 1\}$, the network enjoys ∼16× memory saving compared to its 32-bit floating-point counterpart. When further binarize the weights and activations to only 1-bit (i.e., $\{-1, 1\}$), up to 32×

model-size reduction and 58× speedup on CPUs have been achieved (Rastegari et al., 2016), where the matrix multiplication operations are replaced with light-weighted bitwise XNOR operations.

Despite its appealing characteristics, naively binarizing or ternarizing the transformer model for natural language generation results in several accuracy drops or even a total failure in training. It has been observed that the attention layers of the transformer network are difficult to quantize to low bits. Also, the auto-regressive decoding tends to accumulate errors due to quantization. Given the nature of generative language networks that require high-precision output, quantizing both the activations and weights in these models to extreme bit values is non-trivial and has not been explored before.

2.2 Stats-based max-entropy isometric weight quantization

We propose a statistics-based method for weight binarization/ternarization. Particularly, this novel quantization method considers maximizing the entropy of the quantized weights and reducing the gradient mismatch in the backward pass. Previous works (Courbariaux et al., 2016; Bai et al., 2021b; Zhang et al., 2020) are mainly focused on minimizing the $l2$ loss between the quantized weights and the real-valued weights to find the optimal quantization scheme,

$$\alpha^* = \arg\min ||\alpha \hat{\mathbf{W}}_{\mathbf{Q}} - \mathbf{W}_{\mathbf{R}}||_{l2} \tag{6}$$

where $\hat{\mathbf{W}}_{\mathbf{Q}}$ denotes binary/ternary weights and α^* denotes the optimal scaling factor calculated. Despite the broad application and great success of the classic quantization scheme, we found that merely minimizing the $l2$ loss neglects several critical but intractable issues in ultra-low-bit weight quantization: (1) The information entropy of the quantized weights is not considered. Eq. 1 and Eq. 4 calculate the quantized weights to minimize the distance to the real-valued weights, which could lead to imbalanced quantized weight distribution and harm the quantized weights representation capacity. (2) The quantization function Eq. 1 and Eq. 4 are not isometric, meaning that it does not consider the magnitude consistency between the quantized weights and real-valued weights, while we find that magnitude consistency contributes significantly to accurate gradient estimation.

Considering the above two limitations in previous solutions, we are motivated to design a novel quantization function that enhances information entropy and reduces gradient mismatch. To boost the weights representation capability, in information theory, more information is preserved when the quantized weights contain higher entropy:

$$\max_{p_i} \mathcal{H} = -p_i \log(p_i), s.t. \sum_{i=1}^{N} p_i = 1 \tag{7}$$

with p_i denoting the proportion of real-valued weights being quantized to i^{th} quantization level in total N levels. Eq. 7 can be easily solved with a Lagrange multiplier, and the optimal $p_i^* = \frac{1}{N},\ i \in \{1, 2, \ldots, N\}$, suggesting the best quantization scheme to preserve maximum information entropy is to distribute the real-valued weights in all quantization levels as evenly as possible.

For reducing the gradient mismatch, as suggested by the previous binarization work (Liu et al., 2020b), the magnitude difference between the quantized weight and the real-valued weight will greatly influence the gradient scale and a mismatch in magnitude will be amplified in back-propagation and cause gradient vanishing or explosion during training. Thus it is important to ensure the magnitude of real-valued weights and quantized weights are consistent.

Combining two requirements discussed above, we proposed max-entropy isometric weight quantization. In ternarization, it is formulated as

$$\begin{aligned} \mathbf{W}_{\mathbf{T}}^i &= \alpha_{\mathbf{T}} \lfloor \mathrm{Clip}(\frac{\mathbf{W}_{\mathbf{R}}^i - \mu_{\mathbf{T}}}{\alpha_{\mathbf{T}}}, -1, 1) \rceil \\ \text{where } \mu_{\mathbf{T}} &= \overline{\mathbf{W}_{\mathbf{R}}}, \\ \alpha_{\mathbf{T}} &= \frac{4}{3} \cdot \frac{||\mathbf{W}_{\mathbf{R}} - \mu_{\mathbf{T}}||_{l1}}{n_{\mathbf{W}_{\mathbf{R}}}} \end{aligned} \tag{8}$$

Where $\mathbf{W}_{\mathbf{T}}$ and $\mathbf{W}_{\mathbf{R}}$ refer to the ternary weights and real-valued weights, respectively. The rounding function $\lfloor \cdot \rceil$ and $\mathrm{Clip}(\cdot)$ function quantize weights to $\{-1, 0, 1\}$. $\mu_{\mathbf{T}}$ is the mean of real-valued weights and $n_{\mathbf{W}_{\mathbf{R}}}$ denotes the number of weights in the weight matrix. Scaling factor α is calculated from the weight statistics and follows the entropy rule to scale the real-valued weight $\mathbf{W}_{\mathbf{R}}$ to be evenly distributed in quantization levels. In the ternary case, the weights are quantized to $\{-\alpha_{\mathbf{T}}, 0, \alpha_{\mathbf{T}}\}$. When the real-valued weights are initialized as uniformly and symmetrically distributed (He et al., 2015; Glorot and Bengio, 2010), the scaling factor $\alpha_{\mathbf{T}}$ will distribute $\frac{\mathbf{W}_{\mathbf{R}}^i}{\alpha_{\mathbf{T}}}$ to $[-1.5, 1.5]$, such that the output ternary weights

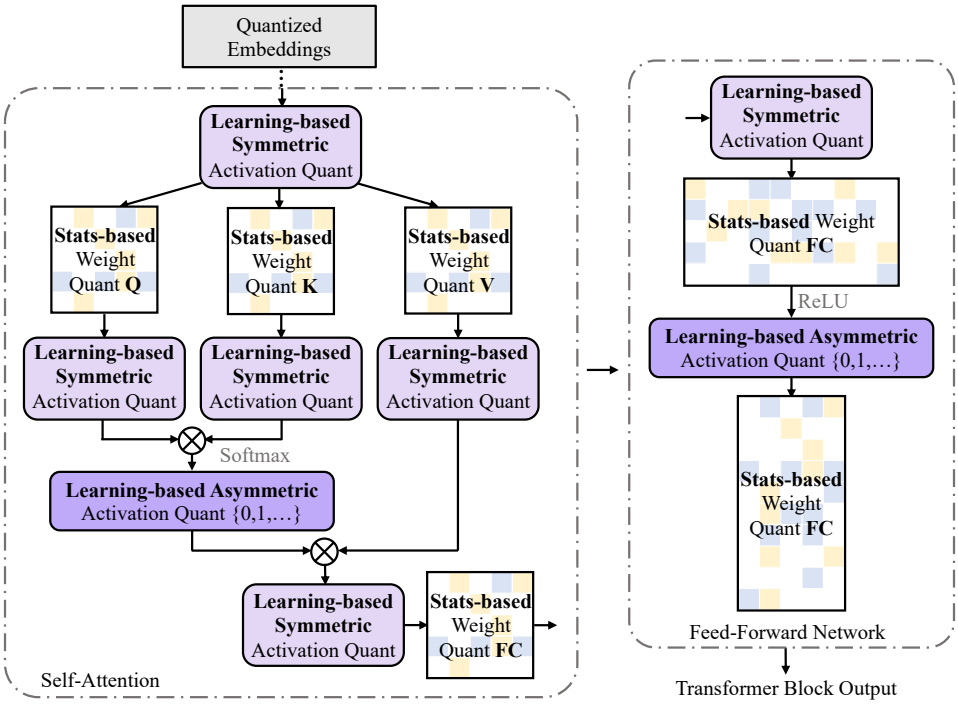

Figure 1: Overview of TBT. A transformer block contains the multi-head self-attention and feed-forward network. We propose a statistic-based quantization method for weights ternarization/binarization and adopt a learning-based asymmetric quantization method for activation in ReLU/Softmax output ($\mathbf{X} \in \mathbb{R}_+^n$) and learning-based asymmetric quantization method for activations that contain both positive and negative values in other layers ($\mathbf{X} \in \mathbb{R}^n$).

will have near uniform distribution in three ternary levels. Meanwhile, Eq. 8 is an isometric mapping where the real-valued weights are scaled by $\frac{1}{\alpha_\mathbf{T}}$ to near [-1, 1] and time $\alpha_\mathbf{T}$ to scale back after quantization. In this way, the magnitude is preserved.

Correspondingly, in the binary case we have,

$$\begin{aligned} \mathbf{W}_\mathbf{B}^i &= \alpha_\mathbf{B} \cdot \text{Sign}(\frac{\mathbf{W}_\mathbf{R}^\text{i} - \mu_\mathbf{B}}{\alpha_\mathbf{B}}) \\ \text{where } \mu_\mathbf{B} &= \overline{\mathbf{W}_\mathbf{R}}, \\ \alpha_\mathbf{B} &= \frac{||\mathbf{W}_\mathbf{R} - \mu_\mathbf{B}||_{l1}}{n_{\mathbf{W}_\mathbf{R}}} \end{aligned} \tag{9}$$

Here $\mathbf{W}_\mathbf{B}$ denotes the binary weights, where substracting the average $\mu_\mathbf{B}$ makes the real-valued weight zero-centered before binarization and thus encourages an even distribution in binarized weights. Then the scaling factor $\alpha_\mathbf{B}$ matches the magnitude between real-valued and binary weights. Particularly, in Eq. 9, $\mathbf{W}_\mathbf{B}^i = \alpha_\mathbf{B} \cdot \text{Sign}(\frac{\mathbf{W}_\mathbf{R}^\text{i} - \mu_\mathbf{B}}{\alpha_\mathbf{B}}) = \alpha_\mathbf{B} \cdot \text{Sign}(\mathbf{W}_\mathbf{R}^\text{i} - \mu_\mathbf{B})$, we explicitly include the $\alpha_\mathbf{B}$ in the denominator to keep the binarization function isometric and the gradients *w.r.t.* weights can be calculated straightforwardly as:

$$\frac{\partial \mathbf{W}_\mathbf{B}^i}{\partial \mathbf{W}_\mathbf{R}^i} \overset{STE}{\approx} \mathbf{1}_{|\frac{\mathbf{W}_\mathbf{R}^i - \mu_\mathbf{B}}{\alpha_\mathbf{B}}| < 1} \tag{10}$$

STE is abbreviated for straight-through estimator (Bengio et al., 2013), which replaces the non-differentiable Sign function with Clip function in the backward pass. We show that the proposed max-entropy isometric weight quantization improves the accuracy of weight binarization / ternarization by 6.0 / 11.53 RougeL scores on the CNN/DailyMail benchmark, respectively. More details can be found in Sec. 3.2.

2.3 Learning-based activation quantization

In contrast to neural network weights that are stored on the disk, activations are calculated on-the-fly. The distribution of activations in a particular layer depends on the network weights as well as the corresponding input sequence, and thus varies from batch to batch. In order to have the quantization function better capture the underlying activation distribution, we propose learning-based activation quantization.

Inspired by BiT (Liu et al., 2022), we divide the activation layers into two categories: the activation

layers with non-negative values ($\mathbf{X_R} \in \mathbb{R}_+$), *i.e.*, Softmax/ReLU layer outputs and the rest of the layers with both positive and negative activations ($\mathbf{X_R} \in \mathbb{R}$). We binarize / ternarize the first activation category ($\mathbf{X_R} \in \mathbb{R}_+$) to $\{0, \alpha\}$ / $\{0, \alpha, 2\alpha\}$, and symmetrically quantize the later activation category ($\mathbf{X_R} \in \mathbb{R}$) to $\{-\alpha, \alpha\}$ and $\{-\alpha, 0, \alpha\}$ in binary and ternary cases respectively. In this way, the activation distribution matches the original full-precision activations and thus reduces the quantization error. Further, we learn to scale the real-valued activations to better fit quantization thresholds, and this learnable scaling factor can be updated end-to-end with the gradients from the network loss to better account for overall network optimization.

In the ternary case, we propose the elastic ternarization function formulated as,

$$\begin{aligned}\mathbf{X}_\mathbf{T}^i &= \alpha_\mathbf{T} \hat{\mathbf{X}}_\mathbf{T}^i \\ &= \begin{cases} \alpha_\mathbf{T} \lfloor \mathrm{Clip}(\frac{\mathbf{X}_\mathbf{R}^i}{\alpha_\mathbf{T}}, 0, 2) \rceil, & \text{if } \mathbf{X_R} \in \mathbb{R}_+ \\ \alpha_\mathbf{T} \lfloor \mathrm{Clip}(\frac{\mathbf{X}_\mathbf{R}'^i}{\alpha_\mathbf{T}}, -1, 1) \rceil, & \text{if } \mathbf{X_R} \in \mathbb{R} \end{cases} \end{aligned} \tag{11}$$

where $\mathbf{X_R}$ and $\mathbf{X_T}$ denote real-valued and ternary activations, respectively. To keep the formula concise, we set $\mathbf{X}'_\mathbf{R} = \mathbf{X_R} - \overline{\mathbf{X_R}}$, denoting the zero-mean real-valued activations. $\alpha_\mathbf{T}$ is the scaling factor. Different from the weight quantization, the scaling factor in Eq. 11 is learned with the gradient update. We follow the practice in (Zhou et al., 2016; Esser et al., 2019) to calculate the gradients with straight-through estimation (STE) bypassing the non-differentiable rounding function:

$$\begin{aligned}&\frac{\partial \mathbf{X}_\mathbf{T}^i}{\partial \alpha_\mathbf{T}} \overset{STE}{\approx} \\ &\quad \begin{cases} \hat{\mathbf{X}}_\mathbf{T}^i - \frac{\mathbf{X}_\mathbf{R}^i}{\alpha_\mathbf{T}} \cdot \mathbf{1}_{0 \leqslant \mathbf{X}_\mathbf{R}^i \leqslant 2\alpha_\mathbf{T}}, & \text{if } \mathbf{X_R} \in \mathbb{R}_+ \\ \hat{\mathbf{X}}_\mathbf{T}^i - \frac{\mathbf{X}_\mathbf{R}'^i}{\alpha_\mathbf{T}} \cdot \mathbf{1}_{|\mathbf{X}_\mathbf{R}'^i| \leqslant \alpha_\mathbf{T}}, & \text{if } \mathbf{X_R} \in \mathbb{R} \end{cases} \end{aligned} \tag{12}$$

The learnable scaling factor can dynamically adapt to different activation distributions and improve the ternarization accuracy. In the binary case, it is formulated as.

$$\begin{aligned}\mathbf{X}_\mathbf{B}^i &= \alpha_\mathbf{B} \hat{\mathbf{X}}_\mathbf{B}^i \\ &= \begin{cases} \alpha_\mathbf{B} \lfloor \mathrm{Clip}(\frac{\mathbf{X}_\mathbf{R}^i}{\alpha_\mathbf{B}}, 0, 1) \rceil, & \text{if } \mathbf{X_R} \in \mathbb{R}_+ \\ \alpha_\mathbf{B} \cdot \mathrm{Sign}(\frac{\mathbf{X}_\mathbf{R}'^i}{\alpha_\mathbf{B}}), & \text{if } \mathbf{X_R} \in \mathbb{R} \end{cases} \end{aligned} \tag{13}$$

Here $\mathbf{X_B}$ denotes the binary activations.

Correspondingly, the gradients *w.r.t.* the scaling factor α can be easily calculated as

$$\begin{aligned}&\frac{\partial \mathbf{X}_\mathbf{B}^i}{\partial \alpha_\mathbf{B}} \overset{STE}{\approx} \\ &\quad \begin{cases} \hat{\mathbf{X}}_\mathbf{B}^i - \frac{\mathbf{X}_\mathbf{R}^i}{\alpha_\mathbf{B}} \cdot \mathbf{1}_{0 \leqslant \mathbf{X}_\mathbf{R}^i \leqslant \alpha_\mathbf{B}}, & \text{if } \mathbf{X_R} \in \mathbb{R}_+ \\ \mathrm{Sign}(\mathbf{X}_\mathbf{R}'^i), & \text{if } \mathbf{X_R} \in \mathbb{R} \end{cases} \end{aligned} \tag{14}$$

We demonstrate that with the learning-based activation quantization method and statistics-based weight quantization scheme, the proposed TBT for the first time is able to quantize the BART model for natural language generation tasks to ternary and even binary weights and activations, and achieve reasonable accuracy on summarization and translation benchmarks.

3 Experiments

In this section, we evaluate the effectiveness of our low-bit quantization scheme for natural language generative model on text summarization benchmarks: CNN/DailyMail (Nallapati et al., 2016) and XSUM (Narayan et al., 2018). We additionally experiment on the machine translation task with mBART on WMT16 English-Romanian (En-Ro) dataset (Bojar et al., 2016a).

3.1 Experimental settings

We follow recent work (Li et al., 2022) in training the quantized network with initialization and knowledge distillation from a full-precision pre-trained model. Specifically, we use the BART-base (Lewis et al., 2019) as our full-precision baseline for summarization tasks and mBART-large (Liu et al., 2020a) for the translation task. We train the quantized models for 20 epochs on 8 GPUs with a batch size of 128 and a learning rate of 2.5e-4 for 8-bit activation models and 5e-4 for binary and ternary activation models.

3.2 Summarization

For the summarization task, we adopt the following benchmarks:

The XSUM dataset (Narayan et al., 2018) consists of 226k documents sampled from the online news website of BBC, together with short, one sentence summaries. Since the summaries are very short, abstractive methods tend to do better on this dataset.

Table 1: Comparison of quantization methods for text summarization on XSUM and CNN/DailyMail benchmarks. We use the "E-W-A (#bits)" notation referring to the number of bits of embeddings, weights and activations, (specifically, 1 denotes binary, 2 denotes ternary). The results of QuantBart, DQ-BART and BlockPruning are quoted from their paper. Additionally, we implement the algorithm developed in BinaryBert, BiBert and TernaryBert to the BART model and report the results, denoted with *. We use the rouge-{1,2,L} as evaluation metrics.

Method	#Bits (E-W-A)	Size (MB)	FLOPs	XSUM R1	XSUM R2	XSUM RL	CNN/DailyMail R1	CNN/DailyMail R2	CNN/DailyMail RL
BART	32-32-32	532.0	1×	43.84	20.79	35.71	44.90	22.25	42.09
QuantBart (Tao et al., 2022)	8 - 8 - 8	138.1	–	40.25	17.78	32.70	–	–	–
DQ-BART (Li et al., 2022)	8 - 8 - 8	138.1	–	42.51	19.61	34.61	44.66	21.92	41.86
Ternary									
Baseline (TWN) (Li et al., 2016)	2 - 2 - 8	39.6	0.25×	39.99	17.13	31.99	42.99	20.05	40.18
QuantBart (Tao et al., 2022)	2 - 2 - 8	39.6	0.25×	39.15	16.72	31.72	–	–	–
DQ-BART (Li et al., 2022)	2 - 2 - 8	39.6	0.25×	40.06	17.34	32.46	42.94	20.07	40.13
TBT	2 - 2 - 8	39.6	0.25×	**42.40**	**19.54**	**34.51**	**43.46**	**20.52**	**40.58**
Baseline (TWN) (Li et al., 2016)	2 - 2 - 2	39.6	0.0625×	12.80	1.21	11.4	12.92	0.32	12.42
TernaryBert* (Zhang et al., 2020)	2 - 2 - 2	39.6	0.0625×	14.03	2.23	11.79	10.95	0.52	8.56
TBT	2 - 2 - 2	39.6	0.0625×	**36.21**	**14.38**	**29.07**	**41.03**	**18.18**	**38.30**
Binary									
Baseline (BWN) (Courbariaux et al., 2016)	1 - 1 - 8	23.2	0.125×	1.90	0.01	1.78	2.78	0.08	2.48
BinaryBert* (Bai et al., 2021b)	1 - 1 - 8	23.2	0.125×	39.76	17.05	31.99	40.66	18.52	28.36
BlockPruning (Lagunas et al., 2021)	–	23	–	–	–	–	41.4	18.7	38.4
TBT	1 - 1 - 8	23.2	0.125×	**40.96**	**18.37**	**33.30**	**42.66**	**19.72**	**39.80**
Baseline (BWN) (Courbariaux et al., 2016)	1 - 1 - 1	23.2	0.0156×	1.90	0.01	1.78	2.78	0.08	2.48
BinaryBert* (Bai et al., 2021b)	1 - 1 - 1	23.2	0.0156×	8.13	0.12	7.69	9.80	0.15	8.62
BiBert* (Qin et al., 2021)	1 - 1 - 1	23.2	0.0156×	7.58	0.06	7.54	14.22	0.13	10.06
TBT	1 - 1 - 1	23.2	0.0156×	**31.68**	**11.19**	**25.29**	**35.56**	**11.71**	**33.23**

CNN/DailyMail (Nallapati et al., 2016) is another news summarization benchmark, with longer documents (~30 sentences) and longer, multi-sentence summaries. The dataset contains close to 300k document-summary pairs.

We use BART-base model (Lewis et al., 2019), which is an English-only encoder-decoder transformer with 140 million parameters. We compare using the standard ROUGE-{1,2,l} metrics for this task.

For the ternary weights and 8-bit activations setting, we compare with two state-of-the-art methods QuantBart (Tao et al., 2022) and DQ-BART (Li et al., 2022). For the fully ternary setting, and the binary quantization experiments, there is no prior art. Therefore we provide a naive quantization baseline, using popular implementations from previous work (Li et al., 2016; Courbariaux et al., 2016), and adapt the binary and ternary methods proposed for the BERT models (Bai et al., 2021b; Qin et al., 2021; Zhang et al., 2020) to BART.

Our main results are summarized in Table 1. In the ternary weights and 8-bit activations setting, TBT improves previous SoTA by up to **2.3 points** in ROUGE score on XSUM, and up to **0.5 points** on CNN/DailyMail. Both improvements are significant.

Further quantizing weights to *binary*, while keeping activations at 8-bit, we are still able to achieve a ROUGE-L score of 33.3 on XSUM, which is 0.8 points higher than the previous *ternary* SoTA (DQ-BART), and comparable on CNN/DailyMail. This is the first demonstration of a binary-weight generative transformer model of competitive accuracy to our knowledge. Additionally, TBT binary weight BART model achieves **1.2 points** higher ROUGE score on CNN compared with the SoTA pruning method with the same compressed model size.

Moving on to ternary and binary activations, there is no prior art, and previous implementations fail to produce meaningful results. Our method, on the other hand, achieves ROUGE-L scores of 29.1 and 38.3 on XSUM and CNN/DailyMail in the fully ternary setting, which are 6.6 and 3.8 points behind the full-precision baseline respectively. Our fully binary (weights and activations) model has a wider gap at 10.4 and 8.9 points, however still manages to produce highly non-trivial output at ROUGE-L scores of 25.3 and 33.2 points for XSUM and CNN/DailyMail.

3.3 Machine translation

We also evaluate our model on machine translation. We adopt the En-Ro benchmark from the

Table 2: Comparison of quantization methods on mBART-large model for translation on WMT16 En-Ro.

Method	#Bits (E-W-A)	Size (GB)	BLEU
mBART (Liu et al., 2020a)	32-32-32	2.44	26.82
DQ-BART (Li et al., 2022)	8 - 8 - 8	0.61	25.91
DQ-BART (Li et al., 2022)	2 - 2 - 8	0.31	23.48
TBT	2 - 2 - 8	0.31	**24.63**
TBT	2 - 2 - 2	0.31	**21.70**
TBT	1 - 1 - 8	0.16	**24.30**
TBT	1 - 1 - 1	0.16	**17.59**

Table 3: Ablation study on the effects of the proposed learning-based activation quantization method and stats-based weight quantization method on XSUM and CNN/DailyMail benchmark.

	Method	#Bits (E-W-A)	XSUM R1	XSUM R2	XSUM RL
1	Baseline (TWN)	2 - 2 - 2	12.80	1.21	11.4
2	+ Activation(learning-based)	2 - 2 - 2	15.05	1.38	12.13
3	+ Weight(stats-based)	2 - 2 - 2	13.79	0.87	12.74
4	+ Both	2 - 2 - 2	**36.21**	**14.38**	**29.07**
5	Baseline (BWN)	1 - 1 - 1	1.90	0.01	1.78
6	+ Activation(learning-based)	1 - 1 - 1	1.90	0.01	1.78
7	+ Weight(stats-based)	1 - 1 - 1	10.96	0.29	10.00
8	+ Both	1 - 1 - 1	**31.68**	**11.19**	**25.29**
			CNN/DailyMail R1	**CNN/DailyMail** R2	**CNN/DailyMail** RL
9	Baseline (TWN)	2 - 2 - 2	12.92	0.32	12.42
10	+ Activation(learning-based)	2 - 2 - 2	13.34	0.99	12.58
11	+ Weight(stats-based)	2 - 2 - 2	19.34	0.42	18.42
12	+ Both	2 - 2 - 2	**41.03**	**18.18**	**38.30**
13	Baseline (BWN)	1 - 1 - 1	2.78	0.08	2.48
14	+ Activation(learning-based)	1 - 1 - 1	2.78	0.08	2.48
15	+ Weight(stats-based)	1 - 1 - 1	15.05	0.35	14.01
16	+ Both	1 - 1 - 1	**35.56**	**11.71**	**33.23**

WMT'16 shared task (Bojar et al., 2016b) to be compatible with previous work. Our base model is an mBART-large model (Liu et al., 2020a), a 680 million parameter multi-lingual encoder-decoder transformer pre-trained on 25 languages.

Table 2 shows our results. In the ternary weight setting with 8-bit activations, we improve the previous SoTA by 1.2 points, achieving 24.63 BLEU. Remarkably our binary weight model also outperforms the previous ternary weight SoTA by almost a full point. It scores 24.3 BLEU – only 1.5 points behind a full mBART model while being 16× smaller.

In the fully ternary and binary settings, where previous methods failed to converge, TBT models are able to reach practical levels of performance, with ternary TBT mBART achieving 21.7 BLEU, and TBT binary mBART at 17.59.

3.4 Ablations

As stated earlier, our main proposed modeling improvement is a combination of two methods: statistics-based quantization for the weights, and learning-based quantization for the activations. We ablate the contribution of these methods and present the results in Table 3.

The results clearly show that while each method can give moderate gains by itself over the baseline, these improvements are not sufficient by themselves to produce meaningful results. None of the ablated models can achieve an R2 score above 1.5. It's only the *combination* of the two, which together stabilize the training and result in good convergence for fully ternary and binary models.

Table 4: Generated average sequence length comparison between baseline method and our method.

Method	#Bits (E-W-A)	XSUM	CNN/DailyMail
BART-base	32-32-32	30.73	99.89
Baseline	2 - 2 - 8	28.53	93.63
TBT	2 - 2 - 8	32.04	95.78
Baseline	2 - 2 - 2	48.41	14.88
TBT	2 - 2 - 2	30.71	88.38
Baseline	1 - 1 - 8	62.0	128.0
TBT	1 - 1 - 8	31.57	97.08
Baseline	1 - 1 - 1	62.0	128.0
TBT	1 - 1 - 1	29.81	67.51

3.5 Sequence length analysis

In language generation tasks, the error compounding issue in the recursive decoder generation process will largely amplify the quantization error or even lead to divergent results, and thus is an harsh factor to test the robustness of a quantization method. The average generated sequence length indicates whether the quantized model can overcome the compounding error and generate reasonable length of text.

In Table 4 we compare the generated sequence length between the proposed method and the baseline method (*i.e.*, TWN (Li et al., 2016) for ternary, BWN (Courbariaux et al., 2016) for binary). Our method successfully produces summarizations with comparable length as the full-precision model on XSUM benchmark, even when both weights and activations are binarized.

Compared to XSUM dataset, for which the document are summarized to only one sentence, CNN/DailyMail is more challenging because it allows longer summary. We can clearly see that, the text generate with our 8-bit activation models can maintain near the similar average length as the full-precision BART model, while the binary and ternary activation models deviate moderately. In contrast, the baseline method is only able to derive

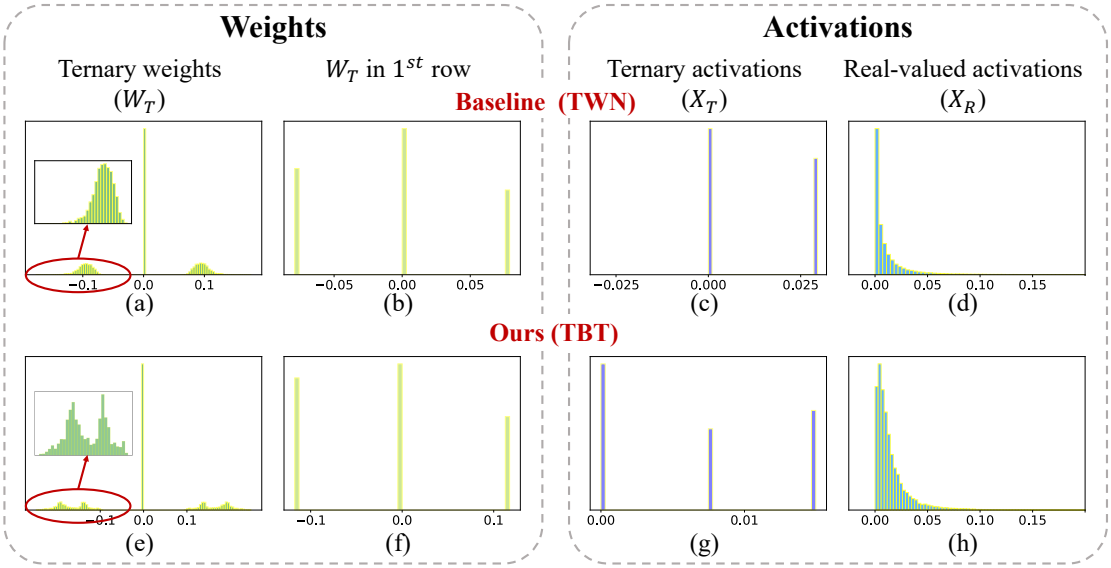

Figure 2: Weight and activation histogram comparison between the baseline TWN method and TBT method for ternarizing BART model on CNN/DailyMail benchmark. The weights are taken from the fully-connected layer of the value matrix in 1^{st} self-attention block in the decoder and activations are the attention outputs of the same layer.

reasonable summarization with 2-bit weight 8-bit activations and fails at lower bit-width, showing the difficult natural of the language generation tasks.

3.6 Visualization

To further understand the effectiveness of the proposed method, we visualize weight and activation histograms in the BART model ternarized with the baseline method and the proposed method in Fig. 2.

Both the baseline method and our method use per-row weight ternarization, and thus a tensor tensor will have #row of scaling factors. As we can see in Fig. 2 (b) and (g), the proposed method allows the weights to be more evenly distributed in three ternarization levels, which can allow higher information entropy in quantized weights, as discussed in Sec. 2.2. Additionally, we calculate the quantized weight distribution entropy (i.e., Eq. 7) in 96 fully-connected layers in the BART-base model and found that the proposed TBT method achieves consistently higher entropy in quantized weights than the baseline method in all the layers. Further, an interesting phenomenon we can see in Fig. 2 (a) (e) is that ternary weights in a baseline model are very close to the Gaussian distribution, in contrast, weights ternarized with TBT are capturing a more sophisticated distribution. This phenomenon implies that the proposed method helps the weights learn more informative patterns and thus better satisfy the high demand for language generation tasks.

For activation quantization, it is evident that the attention layer and the SoftMax output only contain the positive activations ($\mathbf{X_R} \in \mathbb{R}_+$). If simply ternarized to $\{-\alpha, 0, \alpha\}$, the ternary activations will waste one representative level (Fig. 2(d)) and therefore lead to lower accuracy. Instead, the proposed method uses a two-set ternarization method that ternarizes the non-negative activation layer ($\mathbf{X_R} \in \mathbb{R}_+$) to $\{0, \alpha, 2\alpha\}$, and learns the scaling factor α to better fit the underlying real-valued distribution. This ternarization method greatly reduces information loss and enhances the final accuracy.

4 Related Work

Quantization has long been studied to make neural networks more efficient (see (Hubara et al., 2017) for a survey). Due to the popularity of BERT, numerous works have studied quantization for transformer models, starting with 8-bit quantization (Zafrir et al., 2019; Fan et al., 2020), and progressing to 4-bit (Shen et al., 2020; Zadeh et al., 2020), ternary (Zhang et al., 2020) and binary Bai et al. (2021b); Qin et al. (2021); Liu et al. (2022). All of these works have focused on the encoder-only setting.

In the generative setting, Prato et al. (2019); Behnke et al. (2021) demonstrate quantized models for machine translation, and Fan et al. (2020); Bai et al. (2021a) for language modeling, though only for moderate quantization levels (4-8 bits). Most recently, Tao et al. (2022) and Li et al. (2022) pushed weight quantization down to 2 bits (with

8-bit activation quantization) and evaluated on language modeling and summarization. However, our method outperforms these works substantially, while also demonstrating accurate generative transformers with both weights and activations quantized to 2-bit and even 1-bit for the first time.

5 Conclusion

We have demonstrated high accuracy ternary and binary natural language generation models based on a pre-trained transformer encoder-decoder backbone. Quantizing both the weights and the activations of the network allow these models to run on special-purpose hardware using binary and ternary arithmetic, which doesn't require multiplication modules. Therefore our results promise multiple orders of magnitude gains in efficiency while running these models, and can drastically expand the use cases of such models beyond just high end gpu servers. We are especially excited about the implications of our results for larger text generation models such as GPT-3 (Brown et al., 2020). These models have both demonstrated impressive capabilities, while also presenting enormous scaling and computational challenges. Low-bit quantization is a promising approach to mitigate some of these issues. Whether our approach will scale to these models is an open problem and an exciting future research direction.

6 Limitations

We conduct experiments on public datasets of finite sentence length, while generalizability to extremely long sequences or even streaming data has not been verified. Furthermore, the generalizability of the proposed quantization method to other tasks, including computer vision or speech recognition, remains to be tested. In addition, binarization and ternarization require bit-packing to have actual memory savings and dedicated hardware support for real-time acceleration, which is more of a hardware implementation aspect and not studied in this paper.

7 Ethics Statement

We affirm that we contribute to society, avoid harm, and are honest and trustworthy. We respect previous work and appropriately cite the methods and datasets we are using. All data we use is public and no private data is involved. There is some potential risk if the translation technique is maliciously used by a third party and thus we are committed to maintaining the compression techniques we have developed and the general summarization/machine translation techniques used correctly without incurring any form of discrimination.

References

Haoli Bai, Lu Hou, Lifeng Shang, Xin Jiang, Irwin King, and Michael R Lyu. 2021a. Towards efficient post-training quantization of pre-trained language models. *arXiv preprint arXiv:2109.15082*.

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael R Lyu, and Irwin King. 2021b. Binarybert: Pushing the limit of bert quantization. In *ACL/IJCNLP (1)*.

Maximiliana Behnke, Nikolay Bogoychev, Alham Fikri Aji, Kenneth Heafield, Graeme Nail, Qianqian Zhu, Svetlana Tchistiakova, Jelmer Van der Linde, Pinzhen Chen, Sidharth Kashyap, et al. 2021. Efficient machine translation with model pruning and quantization. In *Proceedings of the Sixth Conference on Machine Translation*, pages 775–780.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016a. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016b. Results of the wmt16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. 2019. Learned step size quantization. In *International Conference on Learning Representations*.

Angela Fan, Pierre Stock, Benjamin Graham, Edouard Grave, Rémi Gribonval, Herve Jegou, and Armand Joulin. 2020. Training with quantization noise for extreme model compression. *arXiv preprint arXiv:2004.07320*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2017. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1):6869–6898.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M Rush. 2021. Block pruning for faster transformers. *arXiv preprint arXiv:2109.04838*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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*, pages 7871–7880.

Fengfu Li, Bo Zhang, and Bin Liu. 2016. Ternary weight networks. *arXiv preprint arXiv:1605.04711*.

Zheng Li, Zijian Wang, Ming Tan, Ramesh Nallapati, Parminder Bhatia, Andrew Arnold, Bing Xiang, and Dan Roth. 2022. Dq-bart: Efficient sequence-to-sequence model via joint distillation and quantization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 203–211.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Zechun Liu, Wenhan Luo, Baoyuan Wu, Xin Yang, Wei Liu, and Kwang-Ting Cheng. 2020b. Bi-real net: Binarizing deep network towards real-network performance. *International Journal of Computer Vision*, 128(1):202–219.

Zechun Liu, Barlas Oguz, Aasish Pappu, Lin Xiao, Scott Yih, Meng Li, Raghuraman Krishnamoorthi, and Yashar Mehdad. 2022. Bit: Robustly binarized multi-distilled transformer. *arXiv preprint arXiv:2205.13016*.

Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. 2018. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European conference on computer vision (ECCV)*, pages 722–737.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

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*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. 2019. Fully quantized transformer for machine translation. *arXiv preprint arXiv:1910.10485*.

Haotong Qin, Yifu Ding, Mingyuan Zhang, YAN Qinghua, Aishan Liu, Qingqing Dang, Ziwei Liu, and Xianglong Liu. 2021. Bibert: Accurate fully binarized bert. In *International Conference on Learning Representations*.

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

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8815–8821.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022. Compression of generative pre-trained language models via quantization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4821–4836.

Ali Hadi Zadeh, Isak Edo, Omar Mohamed Awad, and Andreas Moshovos. 2020. Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 811–824. IEEE.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pages 36–39. IEEE.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020. Ternarybert: Distillation-aware ultra-low bit BERT. In *EMNLP*.

Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2016. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 6

☑ A2. Did you discuss any potential risks of your work?
Section 7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

section 4

☑ B1. Did you cite the creators of artifacts you used?
section 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
section 4

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
section 4

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
section 3

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
section 3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
section 3

C ☑ Did you run computational experiments?

section 3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
section 3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
section 3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
section 3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
section 3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.