On the Way to Lossless Compression of Language Transformers: Exploring Cross-Domain Properties of Quantization

Nikita Martynov[1,3], Aleksei Goncharov[1], Gleb Kumichev[1], Evgeniy Egorov[1], Stanislav Pavlov[2,4,5], Mikhail Durinov[2], Aleksandr Zuev[2] and Egor Filimonov[2]

[1]MIL Team, Erevan, Armenia
[2]Huawei Russian Research Institute, Nizhny Novgorod, Russia
[3]SaluteDevices, Moscow, Russia
[4]HSE University, Nizhny Novgorod, Russia
[5]Volga State University of Water Transport, Nizhny Novgorod, Russia
nikita.martynov.98@list.ru, alex.goncharov@mil-team.com, iamglebk@gmail.com,
eugolak@gmail.com, PavlovStanislav@mail.ru, msdurinov@gmail.com,
ya.al.zuev@yandex.ru, filimonov.egor1@huawei.com

Abstract

Modern Transformers achieved impressive results on various Natural Language Processing tasks over the last few years. The one downside of this success is the size of these models. Huge capacity, which sometimes surpasses billions of parameters, improves generalization abilities, but makes it difficult to employ. Developing field of model compression seeks to reduce the model size and inference latency. This research focuses on one of the compression techniques — Post-Training Quantization. We present a methodology to effectively quantize at least 95% of Transformer weights and corresponding activations to INT8 without any access to task-specific data so the drop in performance does not exceed 0.02%. Furthermore, we provide intriguing observations that reflect cross-domain nature of some of the quantization properties.

Keywords: Quantization, Compression, Transformer, BERT

1. Introduction

Natural Language Processing (NLP) enables extracting meaning and learning from text data and achieved impressive progress in recent years. This evolution is attributed to the continually evolving models and techniques in NLP. However, with growth comes the challenge of expanding model sizes, with some models now encompassing billions of parameters as of this writing (Zhao et al., 2023). This rapid growth in model dimensions brings deployment challenges. Emerging domain of model compression aims to tackle these by streamlining model sizes, making it feasible for NLP models to run efficiently on devices like smartphones or basic CPUs (Gupta and Agrawal, 2022).

Popular approaches in this area include distillation, quantization and pruning (Gou et al., 2021; Liang et al., 2021; Gholami et al., 2021). Pruning selectively trims certain parameters or parameter groups from a neural network without significant performance loss (Frankle and Carbin, 2018), whereas distillation involves a smaller neural network (the student) learning in tandem with a pretrained, larger network (the teacher) (Hinton et al., 2015). The strategies in quantization are primarily categorized into Quantization-Aware Training (QAT) (Jacob et al., 2018) and Post-Training Quantization (PTQ) (Lee et al., 2022). While QAT often results in superior performance, it demands additional computational resources and intricate expertise for successful deployment. In contrast, PTQ offers higher speed of obtaining a quantized model, though potentially compromising optimal performance. PTQ encompasses both static and dynamic approaches. In the static method, quantization parameters are determined after training, relying on a representative calibration dataset, with both weights and activations undergoing quantization. Calibration dataset may be unavailable and zero-shot quantization methods, which bypass this need, often employ various heuristics, but tend to underperform, particularly at ultra-low precision levels. On the other hand, dynamic quantization calculates these parameters in real-time during inference for each individual sample, focusing mainly on activations. Both methods seek to balance model accuracy with computational efficiency, and the selection between them is influenced by the specific needs of the application and hardware limitations.

In this study, we describe a quantization approach that compresses a minimum of 95% of model parameters and activations while maintaining marginal degradation in performance, when fine-tune dataset is not available. We gradually design the procedure that employs static PTQ on layers specifically selected so that the quantization has least impact on the model's performance. The quantization parameters calibration is executed with samples obtained from publicly available cor-

LREC-COLING 2024, pages 12435–12442
20-25 May, 2024.

pora. Additionally, we refine the estimate of quantization scaling, drawing from the inherent knowledge of the GeLU (Hendrycks and Gimpel, 2016) activation function. We test the proposed approach through a series of experiments with a number of encoder-based architectures, tasks and datasets.

The reminder is organized as follows. In Section 2 we overview prior works in the field of model compression. Section 3 introduces proposed methodology, and Section 4 describes experimental setup, tasks, datasets and models. Major results are reflected in Section 5 and Section 6 and the work is concluded with a discussion of promising venues for adopting quantization in NLP in Section 7.

2. Related Works

In the landscape of neural network compression, quantization has emerged as a prominent technique to minimize computational and memory overheads. While many studies aim to lower model parameter precision for easier deployment on limited-resource devices, this often leads to a decrease in accuracy, which is considered a crucial challenge in this research.

The one group of studies represented by works (Junczys-Dowmunt et al., 2018; Bhandare et al., 2019; Zafrir et al., 2019; Shen et al., 2020; Kim et al., 2021; Stock et al., 2020; Darvish Rouhani et al., 2020) adopted various forms of QAT when quantizing Transformer (Vaswani et al., 2017) based architectures. These research mainly explore applications of QAT employing PTQ methods as inferior baselines. Among multiple challenges in QAT the problem of overcoming outliers' influence seems to be crucial. The papers (Zafrir et al., 2019; Bondarenko et al., 2021; Wei et al., 2022; Bondarenko et al., 2023; Xiao et al., 2023) focus on providing the corresponding solutions with different compression-accuracy trade-offs.

ZeroQuant (Yao et al., 2022) is a PTQ approach tailored for large Transformer (Vaswani et al., 2017) models combined with a layer-by-layer knowledge distillation algorithm. Empirical results demonstrate that ZeroQuant can effectively reduce precision to INT8 for models like BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020) with minimal accuracy loss. (Yvinec et al., 2023) provides REx — a data-free post-training quantization method, leveraging residual error expansion and group sparsity.

Alternative methods encompass the Hessian Aware Quantization technique (Shen et al., 2020) and approximate second-order strategies adapted for quantizing large language models (Frantar et al., 2023).

This research emphasizes static PTQ, adapting ZeroQ (Cai et al., 2020) method originally applied in Computer Vision (CV) to effectively select layers that undergo compression. We also address the problem of complete unavailability of fine-tune datasets by leveraging portions of open-source data as it is proposed in (Yu et al., 2021) for the domain of CV.

3. Methodology

In this work we design quantization procedure so that the drop in performance does not surpass 0.02% while fine-tune dataset is not available. We employ static PTQ on specifically selected layers exploiting open-source data to calibrate quantization parameters. Further in this chapter, we disclose the details on formulas and algorithms behind terms used to describe the final quantization algorithm.

3.1. Static Post Training Quantization

In static PTQ, the quantization operation Q for floating point tensor x is defined as follows:

$$Q(x) = round\left(\frac{x}{s} + z\right) \tag{1}$$

The parameters s and z are scale and zero-point. The goal of quantization method is to define s and z. Static PTQ works only after the model training is complete. The procedure includes following stages:

- Calibration. During this stage observers are attached to selected layers and activations. The observers analyze the distribution of values in target tensors and estimate the parameters s and z.

- Conversion. On the conversion stage all the target tensors are converted to integer values according to the equation 1.

3.2. Layers selection

We quantize 95% of model's weights and activations to INT8 and the rest is left in the original precision, which means the layers that undergo compression still have to be carefully chosen concerning the potential drop in performance. To address the challenge of layers selection we adapt corresponding algorithm that has been originally introduced in CV. ZeroQ (Cai et al., 2020) supports both uniform and mixed-precision quantization. For the latter, the framework proposes a novel Pareto frontier based method to automatically determine the mixed-precision bit setting for all layers, with no manual search involved. We construct a Pareto frontier to show how sensitive a certain layer of a neural network is to the quantization procedure by calculating the Kullback-Leibler divergence between the output logits of the original model and its partially quantized counterpart. We then employ it to determine the optimal combination of layers for compression regarding their sensitivity to the quantization.

3.3. Calibration Data

PTQ methods need access to original fine-tune dataset to calibrate s and z parameters for a particular task. This is often not possible due to privacy and security concerns.

We found that a dataset used in unsupervised pre-training of the model can be employed to accurately estimate layers' sensitivity to the quantization and hence to configure layer combinations for PTQ as well as to calibrate quantization parameters without access to the original fine-tune corpus. This scheme works under the assumption that the initial weights of the model were trained with one of the openly available datasets. This assumption is fulfilled for the majority of the NLP models and can be adjusted in a handful of cases otherwise.

This study exploits BookCorpus dataset (Zhu et al., 2015) since it has been used in pre-train procedures of all models listed in Section 4.2.

3.4. Data-Independent Estimation

Estimate of the quantization parameters for PTQ involves calculation of minimum and maximum values of the activations. Wide range of Transformer (Vaswani et al., 2017) models including those in Section 4.2 use GeLU (Hendrycks and Gimpel, 2016) activation function on some of their layers. The GeLU (Hendrycks and Gimpel, 2016) function has a defined absolute minimum, which means the activation values do not fall below this minimum. This allows one to make a better data-independent estimate for the quantization offset parameter z by calculating the lower bound for the activations. Empirical results show that by adjusting procedures that estimate the quantization region, we can improve the quality of the quantized model.

4. Experiments

In this section, we describe a comprehensive set of experiments, which test the quantization algorithm performance relative to the most common tasks required of NLP models.

4.1. Tasks and Datasets

The quantization process inevitably introduces an activation error rate throughout the neural network. In order to measure the influence of error rate on the practical tasks, the following suit of tasks and corresponding datasets is selected. This set covers the mainstream tasks for language models and includes benchmarks that have been proven as quality measure for such models.

Model	Size	Params
BERT	440	110M
RoBERTa	501	125M
TinyBERT	266	67M
XLNet	467	110M

Table 1: Number of parameters and size of the models selected for testing proposed methodology. *Model* refers to the type of architecture, *Size* describes the size of a model in Megabytes and *Params* accounts for the number of parameters in a model, where *M* stands for million.

Reading Comprehension. To solve reading comprehension task for language model means to provide an answer to a question given a text that supposedly contains that answer. To evaluate models' performance in this task we choose RACE (ReAding Comprehension dataset from Examinations) (Lai et al., 2017), which is a dataset specifically designed for testing human reading skills. By assessing performance on RACE (Lai et al., 2017) we also aim to analyze reasoning abilities of the quantized models, since examples, which require reasoning skills, constitute a significant part of the dataset.

Text Classification. In classification tasks we examine quantized models' capacity to jointly extract information from the texts and grasp its emotional palette. The latter is greatly highlighted in sentiment classification datasets. For binary task we choose IMDB dataset (Maas et al., 2011) and for multi-label classification we employ Amazon-2 (He and McAuley, 2016), which is an updated version of the Amazon review dataset (Zhang et al., 2015) released in 2014. For the latter, it is usually used to build recommender system, however, for text classification task we acquire only review texts and their ratings.

Natural Language Inference. Natural language inference (NLI) is the task of determining whether a hypothesis is true (entailment), false (contradiction), or undetermined (neutral) given a premise. We selected QNLI (Question-answering NLI) (Wang et al., 2018) and MNLI (The Multi-genre NLI) (Williams et al., 2018) datasets to ensure the variety of text sources and test the quantized models' potential to generalize accross multiple domains. QNLI (Wang et al., 2018) is derived from the Stanford Question Answering Dataset v1.1 (SQuAD v1.1) (Rajpurkar et al., 2016), while MNLI (Williams et al., 2018) offers ten distinct genres (Face-to-face, Telephone, 9/11, Travel, Letters, Oxford University Press, Slate, Verbatim, Goverment and Fiction) of written and spoken English data.

Dataset	BERT			RoBERTa			TinyBERT			XLNet		
	95%	0%	Δ	95%	0%	Δ	95%	0%	Δ	95%	0%	Δ
RACE	65.93	65.91	**0.02**	72.86	72.86	0.00	59.27	59.27	0.00	66.34	66.35	-0.01
IMDB	92.65	92.64	**0.01**	94.33	94.35	-0.02	89.66	89.67	-0.01	93.94	93.94	0.00
Amazon-2	95.85	95.85	0.00	96.63	96.64	-0.01	96.15	96.15	0.0	96.48	96.48	0.00
QNLI	88.38	88.39	-0.01	91.95	91.96	-0.01	87.74	87.72	**0.02**	89.18	89.18	0.00
MNLI	81.08	81.10	-0.02	87.35	87.34	**0.01**	81.41	81.42	-0.01	86.26	86.27	-0.01

Table 2: The results of the models on different test sets of datasets listed in Section 4.1. We report the average *Accuracy* of five consecutive runs. *95%* stands for performance of the fine-tuned model with at least 95% of weights and activations quantized to INT8 according to the proposed methodology, *0%* refers to the fine-tuned unquantized model, Δ shows difference between obtained results, which is calculated as performance of quantized model minus performance of full model. Positive differences indicate *Accuracy* of the quantized model is higher.

4.2. Models

To test the proposed methodology we use a number of encoder-based architectures. The choice of encoder-based models is decided due to limitations of computational resources.

BERT (Devlin et al., 2019) We use base uncased pre-trained checkpoint from the official GitHub repository[1].

RoBERTa (Liu et al., 2019) is built upon BERT (Devlin et al., 2019), but with much more careful design of hyperparameters and pre-train procedure. Again we employ base pre-trained checkpoint from model's GitHub repository[2].

TinyBERT (Jiao et al., 2020) The distilled version of BERT(Devlin et al., 2019). We acquire second version of six layer checkpoint with only general distillation from the official GitHub repository[3].

XLNet (Yang et al., 2019) Represents autoregressive style of pre-train procedures. We use base pre-trained checkpoint from official GitHub repository[4].

The sizes of models and corresponding number of parameters are provided in Table 1.

4.3. Procedure

To obtain results that correspond to full unquantized architectures we fine-tune models described in Section 4.2 on each task and dataset mentioned in Section 4.1 separately with *batch size* of 32, *AdamW optimizer* (Loshchilov and Hutter, 2018) with a constant *learning rate* of 3e-05 for ten *epochs* with early stopping criterion, which is estimated on a development subset. The resulting checkpoints are evaluated on corresponding test sets.

We then quantize obtained checkpoints. We use ten randomly sampled mini-batches of 256 samples from BookCorpus (Zhu et al., 2015) to estimate layers' sensitivity to the quantization and build corresponding Pareto frontier. The latter is used to determine the optimal combination of layers to perform quantization on considering the overall number of parameters of the selected layers exceed 95% of the model's weights. We then execute PTQ on the chosen layers. Calibration of z and s is done with the same data employed when selecting optimal combination of layers. The quantized model is evaluated then on test sets of the original fine-tune dataset. The results are provided in Table 2.

5. Results

Table 2 summarizes main results obtained in this research. The received metrics indicate drop in performance does not exceed 0.02%. Moreover, several evaluations report quantized models reach even higher Accuracy score than the original ones. This trend is consistent across all the models but XLNet (Yang et al., 2019). We address this finding to regularization properties of the quantization. Although performance drops fluctuate around zero for all evaluated models, the intrinsic structure of target dataset may still affect the results. All the models but RoBERTa (Liu et al., 2019) suffer decrease in metric on the test set of MNLI (Williams et al., 2018) dataset. We assume this is due to complex nature of source data present in MNLI (Williams et al., 2018).

6. Discussion

The findings obtained offer compelling evidence regarding the efficacy of the proposed method. The algorithm demonstrates the capability to quantize over 95% of a model's weights without requiring access to the original dataset, with a performance degradation of less than 0.02%. The process involves two key stages: the identification of layers

[1] https://github.com/google-research/bert

[2] https://github.com/facebookresearch/fairseq/tree/main/examples/roberta

[3] https://github.com/huawei-noah/Pretrained-Language-Model/tree/master

[4] https://github.com/zihangdai/xlnet

based on their sensitivity to quantization and the subsequent calibration of quantization parameters utilizing data from publicly available corpora. While seemingly straightforward, these steps are underpinned by several assumptions.

Firstly, the method leverages the uneven distribution of information across the layers of the architecture, enabling the quantization of a significant portion of layers with minimal impact on the overall signal fidelity. However, the applicability of this concept is contingent upon the inner workings of a specific model, potentially limiting its transferability to diverse architectural configurations.

Second, the calibration of quantization parameters relies on text samples from open datasets utilized during the model's pre-train phase. Notably, experimental results suggest that fine-tune data may not be imperative for reducing quantization errors. Nonetheless, it remains unclear whether the effectiveness of calibration hinges on the source of the text samples, i.e., whether they must align with the datasets used for pre-train procedure.

These topics reveal the promising venues for the future work for both the refinement of the presented methodology and unveiling properties of quantization in general.

7. Conclusion

To address challenges of model size expansion we formulated a strategy to quantize NLP Transformer (Vaswani et al., 2017) models without compromising their performance. Two central observations stand out in proposed approach. First, we demonstrate that it is possible to achieve high-quality PTQ of pre-trained then fine-tuned models without relying on a fine-tune dataset. Second, we expanded the application of ZeroQ (Cai et al., 2020) method, which was previously exclusive to computer vision models, to NLP tasks. A significant part of methodology involved leveraging the BookCorpus dataset (Zhu et al., 2015) for the quantization process. This approach effectively bypasses the need to access task-specific datasets on which the models were originally fine-tuned. We believe that provided findings open venues for further exploration of simple yet effective PTQ methods and their applicability to NLP models.

8. Limitations

The potential domain shift in NLP models when applied to different datasets remains a concern, with its impact on performance yet to be fully understood. Our method's reliance on the BookCorpus dataset (Zhu et al., 2015) raises questions about its generalizability to other datasets. The approach is tailored to specific NLP model architectures, and its adaptability to other architectures is uncertain. Further research is needed to address these challenges.

9. Ethics Statement

We understand the importance of maintaining ethical principles throughout the course of presented work. We take the following aspects of the research into consideration to appeal to potential ethical implications.

Biases. Transformer (Vaswani et al., 2017) based models employed in this work have been pre-trained on datasets that embody a considerable portion of texts gathered from the Internet. These texts are likely to contain a number of biases that may be passed to pre-trained models. We recognize the possibility of biases to appear in models' predictions. Still thorough evaluation is needed to exploit models' sustainability considering various ethical aspects.

Carbon footprint. Training Transformer (Vaswani et al., 2017) models requires significant portion of compute resources, which inevitably affects the environment due to considerable amount of CO2 emissions (Strubell et al., 2019). In this research we proposed a PTQ procedure that neither relies on additional fine-tuning nor expects extensive pre-training. Further research in the field of model compression may also help to reduce carbon footprint and degrade environmental impact.

10. Bibliographical References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *stat*, 1050:21.

Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2023. Quantizable transformers: Removing outliers by helping attention heads do nothing.

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.

Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178.

Bita Darvish Rouhani, Daniel Lo, Ritchie Zhao, Ming Liu, Jeremy Fowers, Kalin Ovtcharov, Anna Vinogradsky, Sarah Massengill, Lita Yang, Ray Bittner, Alessandro Forin, Haishan Zhu, Taesik Na, Prerak Patel, Shuai Che, Lok Chand Koppaka, XIA SONG, Subhojit Som, Kaustav Das, Saurabh T, Steve Reinhardt, Sitaram Lanka, Eric Chung, and Doug Burger. 2020. Pushing the limits of narrow precision inferencing at cloud scale with microsoft floating point. In *Advances in Neural Information Processing Systems*, volume 33, pages 10271–10281. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Manish Gupta and Puneet Agrawal. 2022. Compression of deep learning models for text: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(4):1–55.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in c++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. In *International conference on machine learning*, pages 5506–5518. PMLR.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Jemin Lee, Misun Yu, Yongin Kwon, and Taeho Kim. 2022. Quantune: Post-training quantization of convolutional neural networks using extreme gradient boosting for fast deployment. *Future Generation Computer Systems*, 132:124–135.

Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. 2021. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

OpenAI. 2023. Chatgpt [large language model].

Hariom A. Pandya and Brijesh S. Bhatt. 2021. Question answering survey: Directions, challenges, datasets, evaluation matrices.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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.

Pierre Stock, Angela Fan, Benjamin Graham, Edouard Grave, Rémi Gribonval, Herve Jegou, and Armand Joulin. 2020. Training with quantization noise for extreme model compression. In *International Conference on Learning Representations*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. 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*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. 2022. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, 35:17402–17414.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. 2020. A survey of deep learning techniques for neural machine translation.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183.

Haichao Yu, Linjie Yang, and Humphrey Shi. 2021. Is in-domain data really needed? a pilot study on cross-domain calibration for network quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3043–3052.

Edouard Yvinec, Arnaud Dapgony, Matthieu Cord, and Kevin Bailly. 2023. Rex: Data-free residual quantization error expansion.

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)*. 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)*. IEEE.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2018. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

11. Language Resource References