On the Distribution, Sparsity, and Inference-time Quantization of Attention Values in Transformers

Tianchu Ji[1], **Shraddhan Jain**[2], **Michael Ferdman**[2],
Peter Milder[1], **H. Andrew Schwartz**[2], **and Niranjan Balasubramanian**[2]

[1,2]Stony Brook University
[1]{tianchu.ji, peter.milder}@stonybrook.edu
[2]{shrjain, mferdman, has, niranjan}@cs.stonybrook.edu

Abstract

How much information do NLP tasks really need from a transformer's attention mechanism at application-time (inference)? From recent work, we know that there is sparsity in transformers and that the floating-points within its computation can be discretized to fewer values with minimal loss to task accuracies. However, this requires retraining or even creating entirely new models, both of which can be expensive and carbon-emitting. Focused on optimizations that do not require training, we systematically study the full range of typical attention values necessary. This informs the design of an inference-time quantization technique using both pruning and log-scaled mapping which produces only a few (e.g. 2^3) unique values. Over the tasks of question answering and sentiment analysis, we find nearly 80% of attention values can be pruned to zeros with minimal ($< 1.0\%$) relative loss in accuracy. We use this pruning technique in conjunction with quantizing the attention values to only a 3-bit format, without retraining, resulting in only a 0.8% accuracy reduction on question answering with fine-tuned RoBERTa.

1 Introduction

While the verdict is still out on which large language model will prove best, at this point in time, all contenders rely on multi-headed attention over multiple layers. Many have investigated whether attention (the output of the softmax, α) itself is *qualitatively* sensible (e.g., correlating with linguistic aspects) (Vig and Belinkov, 2019; Clark et al., 2019; Voita et al., 2018, 2019; Kovaleva et al., 2019; Rogers et al., 2020) or how useful it is for interpreting models (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Brunner et al., 2020; Rogers et al., 2020). Others have focused on inducing sparsity in the attention: whether some of the structural components (the softmax function, attention heads and layers) introduce attention sparsity (Correia et al., 2019; Michel et al., 2019; Voita et al., 2019; Sajjad et al., 2020), if the model tends to focus on a small amount of tokens (Clark et al., 2019; Ramsauer et al., 2020), and the interpretability of such sparsity (Chen et al., 2020; Rogers et al., 2020). Yet, little is known about our ability to induce sparsity or reduce its values *at application-time*, and what role the inherent sparsity could play in building inference-time efficient transformers.

This work focuses on a systematic study of the quantitative distribution of the attention values across the layers and heads as well as the potential for reducing the information content of attention values during inference at application-time[1]. We consider two popular pretrained transformer models: BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) over tasks of Masked Language Modeling as well as question answering and sentiment analysis. We explore the attention distributions on the different models and tasks, and quantitatively profile the sparse attention that commonly exists in the transformer model. Motivated by the high levels of inherent sparsity in these distributions, we design a pruning and quantization technique and test the limits of information necessary from attention.

We find that most attention values can be pruned (i.e. set to zero) and the remaining non-zero values can be mapped to a small number of discrete-levels (i.e. unique values) without any significant impact on accuracy. Approximately 80% of the values can be set to zero without significant impact on the accuracy for QA and sentiment analysis tasks. Further, when we add quantization utilizing a log-scaling, we find a 3-bit discrete representation is sufficient to achieve accuracy within 1% of using the full floating points of the original model.

[1]Our analyzing code and data are available at https://github.com/StonyBrookNLP/spiqa

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4147–4157

2 Method

To analyze attention distribution we first plot histograms of attention values for BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) models. We also compute a *sparsity distribution* using the proportion of the attention values smaller than a given threshold. For attention pruning, we find attention values that are below a specified threshold and replace them with zero. We experiment with different thresholds. For quantization to k-bits we map the continuous attention values to one of 2^k real values[2]. We use two methods: (i) Linear - Bin the attention values to 2^k quantiles and set the mid-point of each as the quantized value. (ii) Log - Bin the log transformed attention values and pick the mid-point of each on the log scale as the quantized value. The quantization methods are explained in detail in Appendix E.

We apply these inference-time (i.e. no training) techniques on three tasks: masked language modeling, question answering and sentiment analysis. For QA we used BERT[3] and RoBERTa[4] models fine-tuned on SQuAD v1.1 (Rajpurkar et al., 2016). For sentiment analysis we used RoBERTa[5] fine-tuned on the SST-2 dataset (Socher et al., 2013). For both these tasks we report accuracy on the corresponding development sets. For the Masked Language Modeling (MLM) task we report pseudo-perplexity (Salazar et al., 2020) computed on the Huggingface Wikipedia dataset[6].

3 Evaluation

Attention distribution and sparsity. A thorough quantitative analysis on the attention distribution could help build efficient transformers by providing useful information, such as the degree of sparsity and the range of the attention values. We plot the histogram of each token's attention to all the others (α_i) and provide three examples of the heads in Figure 1 to investigate the density of the attention values, how differently the tokens attend to others in the same attention head, and how sparse a token/head/layer's attention can be. We find that, for most of the heads, attention forms a lognormal-like distribution similar to Figure 1a. On some heads, some of the attention for query token (α_i) have more tiny attention values (α_{ij}) and induce more sparsity than others (like in Figure 1c). We also observe entire heads with high sparsity, in which nearly all tokens only slightly attend to others (like in Figure 1b). Our observation confirms the existence of sparsity in the attention heads.

A key motivation for us is to quantitatively characterize sparsity, especially in terms of how much potential there is in reducing the information content in attention values. To this end, we specifically measure the proportion of small attention values by counting the number of α_{ij} that sum up to 0.5 in each α_i. This indicates that most heads focus strongly on fewer than 10 tokens on average (details in Appendix A), leading to notable sparsity and suggesting large potential for conveying the same information as continuous attention values using fewer discrete levels.

Beyond these, we occasionally observe outlier attention histograms (like the outliers between $[10^{-4}, 10^{-1}]$ in Figure 1b). We also found noticeable differences on the attention histograms from layer to layer. These findings are related to the works on the syntactic heads/special tokens (Voita et al., 2019; Kovaleva et al., 2019; Voita et al., 2018; Clark et al., 2019; Rogers et al., 2020)) and the differences of the layers/heads (Correia et al., 2019; Clark et al., 2019). We discuss how our findings relate to them in Appendices B and C.

Limited effect of near-zero attention values during inference. The inherent sparsity we observed motivates us to explore the sparsity of attention at inference-time—how much attention can be pruned during inference, without impacting the model accuracy? By setting up a series of pruning thresholds, we clamp different proportions of the attention to zero and examine how attention sparsity affects the accuracy, on both pretrained and fine-tuned models. The results shown in Figure 2 indicate that the sparsity can grow above 80% with only a 0.1%–1.3% drop in accuracy. Specifically, the pretrained BERT model achieves 99.9% of the original performance with 87% of the sparsity on Masked Language Modeling. By comparing RoBERTa's accuracy on different tasks, we find that sentiment analysis suffers more from increased sparsity, suggesting that different models are differentially sensitive to the induced sparsity. Our results quantitatively show how much sparsity can be induced in all the attention values without losing

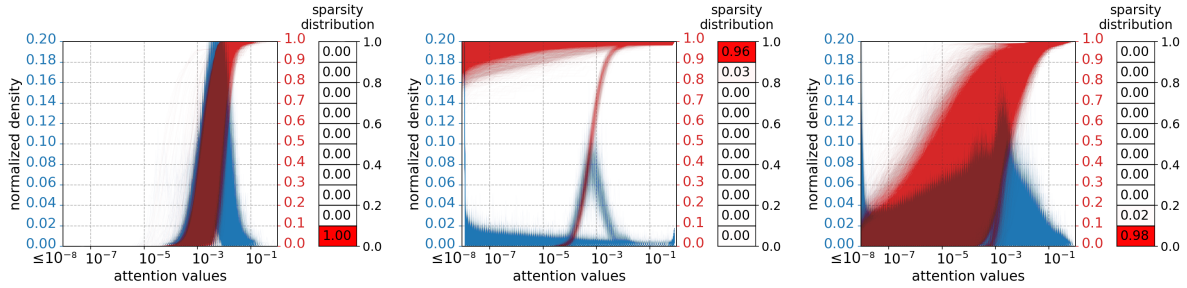
[2] Note here we use full precision floating point rather than a k-bit value since our main goal is to see how many discrete levels of attention is needed.

[3] http://huggingface.co/csarron/bert-base-uncased-squad-v1

[4] http://huggingface.co/csarron/roberta-base-squad-v1

[5] http://huggingface.co/textattack/roberta-base-SST-2

[6] https://huggingface.co/datasets/wikipedia



(a) Layer 1 Head 4 (b) Layer 2 Head 3 (c) Layer 12 Head 11

Figure 1: Normalized histograms (in blue) and cumulative histograms (in red) for every token's attention to others (α_i) at different heads in the pretrained RoBERTa model, starting from 10^{-8}. The histograms show different patterns of attention distribution. E.g., in (b) many tokens' attention form an evenly distributed histogram from 10^{-8} to 1, and most of the α_i have 80%–100% of all the attention values (α_{ij}) $\leq 10^{-8}$. This indicates a higher level of sparsity compared to (a) and (c). The "sparsity distribution" bar on the right shows the density of α_i to each level of sparsity. E.g., the red cell with "0.96" between 0.9–1.0 in (b) means 96% of all α_i have sparsity between 90%–100%, whereas the sparsity is the proportion of α_{ij} in α_i that are less than 10^{-8}.

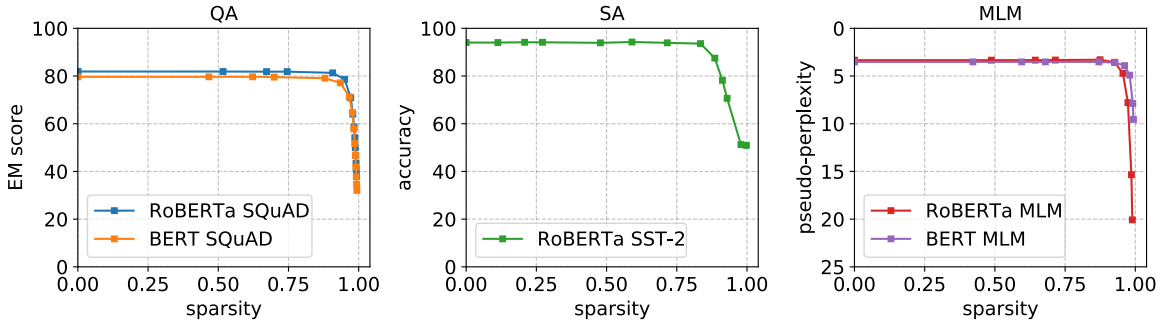
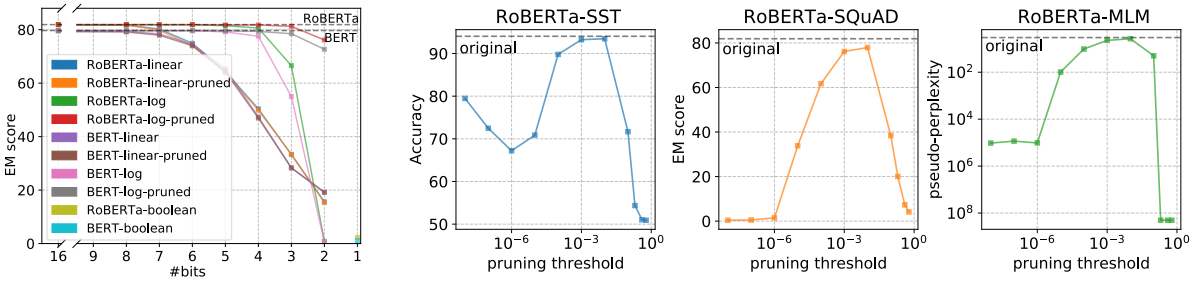


Figure 2: Exact Match score (for QA), Accuracy (for SA) and pseudo-perplexity (for MLM) under different levels of sparsity that we induce, showing that on these models and tasks ∼80% of the sparsity can be induced with limited performance drop. X-axis values denotes the induced sparsity levels measured as the proportion of the attention values less than a specified threshold.

accuracy, suggesting that one can expect to prune up to 80% of the attention values without retraining.

Quantizing pruned attention. Quantization is often used to compress transformer models for higher computational and memory efficiency. Recently Prato et al. (2020) showed that for machine translation, attention values in transformers can be quantized with only a small impact on accuracy. While their results suggest that full precision attention values may not be necessary for high accuracy, it is unclear if one can retain the accuracy in inference-time quantization in general settings i.e., without retraining. Bhandare et al. (2019); Shen et al. (2020); Prato et al. (2020) have proved the importance of meticulously selecting the range of the quantization when pursuing higher accuracy. Intuitively, pruning the tiny attention values will lead to a narrower quantization range with more precise value representatives. For example, if all $\alpha < 10^{-3}$ are pruned before 3-bit quantization, all numbers we need to quantize will land in $[10^{-3}, 1]$ rather than $[0, 1]$, with the 8 quantiles of the quantization located more densely; this forms a higher resolution within the quantization range compared to the non-pruned version. Since we observed that pruning most of the attention values during inference has minimal effect on the accuracy when removing only the tiny attention values ($\alpha < 10^{-3}$ in our case), we hypothesize that properly pruning attention values will help increase the accuracy of the quantized model.

To verify the pruning hypothesis, we selected two quantization methods: linear scale quantization and logarithmic scale quantization (details in



(a) EM scores of the models with differently quantized attention

(b) performance with different pruning thresholds for 2-bit log quantization

Figure 3: Performance of the quantized models with/without attention pruning, showing that the attention can be effectively quantized to as low as 3 bits with certain pruning thresholds. (a) Exact Match scores for the QA with different quantization methods on fine-tuned BERT and RoBERTa. "Boolean" quantization is provided as the extreme case of quantization to a single bit. The pruning has only negligible effect on the linear scale quantization so that "*-linear" and "*-linear-pruned" curves are highly overlapped. (b) Accuracy of the fine-tuned RoBERTa models with 2-bit quantized attention for QA, SA and MLM respectively. The attention is pruned before quantization by using different thresholds (shown on the x-axis). In all the figures, the original model's performance scores are marked with black dashed lines.

Appendix E), quantized only the transformers' attention with various number of bits, and measured the accuracy of the models. Then we repeated the experiment but pruning $\alpha < 10^{-3}$ (which creates $\sim$80% sparsity with limited accuracy drop in our sparsity experiment) before quantizing the attention.

We evaluate the models on different tasks to compare how pruning the attention affects the accuracy when quantizing. Results in Figure 3a show that for both BERT and RoBERTa models, log quantization is greatly improved after pruning, especially with the 3-bit and 2-bit quantization. Notably, the 3-bit log quantization with pruning only loses 0.8% and 1.5% of the original accuracy for the RoBERTa and BERT, respectively. Contrarily, the pruning has very limited effect on the linear quantization because the selected pruning threshold results only in a negligible change to the effective quantization range. (Details are provided in Appendix F.) We also repeated the experiment on other tasks and found 2-bit log quantization with pruning only loses 0.7% accuracy on RoBERTa fine-tuned for sentiment analysis. (Full results are provided in Appendix D.)

We further experimented with different pruning thresholds (Figure 3b) and observed that pruning $\alpha < 10^{-2}$ gives the best performance; the threshold can undermine model accuracy if it is either too large ($> 10^{-2}$) or too small ($< 10^{-3}$).

Our results prove that pruning the sparse attention values helps recover model accuracy with log-scale quantization methods, without any retraining or fine-tuning. With attention pruning, a transformer can retain a comparable amount of accuracy even with a simple, low-precision quantized attention (in our case, a 3-bit log quantization).

Discussion. Sparsifying the attention can help reduce both the computation and memory cost of self-attention during inference. Our experiments above demonstrate that it is possible to prune approximately 80% of attention values while quantizing them to a 3-bit representation. Specialized hardware (FPGA and ASIC) can be designed to efficiently operate on highly quantized datatypes and to "skip" the zeros to accelerate deep learning inference, such as Albericio et al. (2016) (which targets CNNs). Our results show that such an accelerator could effectively reduce the arithmetic cost of computing attention matrices by 80% and reduce the memory footprint of the attention matrices by up to 96% (compounding the effect of sparse representation and quantization). Although attention matrices are not occupying a huge amount of storage, these memory savings can potentially greatly increase the efficiency of a specialized hardware accelerator by reducing its on-chip SRAM usage and/or its memory bandwidth requirement. Further, the computational savings can help reduce the latency. Lastly, it is important to note that the benefits of attention sparsity may extend much further than just computing attention values themselves; other computations in the transformer network can also

benefit from leveraging the high degree of sparsity without retraining/fine-tuning, potentially yielding larger benefits. Future work will investigate the computational benefits of utilizing attention sparsity and the design of customized hardware accelerators to efficiently do so.

4 Related Work

Attention distribution. Many have abstractly studied the attention distribution from different aspects (Clark et al., 2019; Pascual et al., 2021; Ramsauer et al., 2020; Correia et al., 2019), but none specifically have shown the histogram of the α_i directly, nor did they investigate the sparse attention values quantitatively. Correia et al. (2019) indicated that not all of the sparsity in attention was caused by the softmax, and it remained unclear whether such sparsity affected accuracy (which is inspected in this paper).

Pruning. Voita et al. (2019); Sajjad et al. (2020); Michel et al. (2019); Kovaleva et al. (2019) pruned one or more heads/layers resulting in comparable or higher model accuracy, either with or without fine-tuning. These approaches assume that some heads/layers interpret the information redundantly, which is not always true (Brunner et al., 2020; Rogers et al., 2020). In contrast, our work focuses on a more general method of inducing attention sparsity without operating at layer/head granularity.

Quantization. Bhandare et al. (2019); Shen et al. (2020); Prato et al. (2020) have shown benefits from selecting the quantization range, which motivates us to prune the attention before quantization (Section 3). Kim et al. (2021); Zafrir et al. (2019); Prato et al. (2020) required re-training while ours does not. Zhang et al. (2020); Bai et al. (2020); Zadeh et al. (2020) focused on quantizing the weights rather than the attention values, which is out of our scope.

Sparse transformers and attention visualization Parmar et al. (2018); Child et al. (2019); Ho et al. (2019); Beltagy et al. (2020); Ainslie et al. (2020); Li and Chan (2019); Tay et al. (2020) have proposed/summarized various kinds of efficient transformers utilizing induced attention sparsity. However, none of them quantitatively analyzed the statistical distribution and the tiny values of the attention. Vig (2019); Hoover et al. (2020) proposed instance-level attention visualization tools. These are complementary to our quantitative visualization of the distributions of all attention values.

5 Conclusion

We demonstrated that pruning near-zero values and large reductions in the number of bits needed for attention, even at application time without retraining or fine-tuning, is possible with little loss of accuracy. This suggests attention plays a very coarse role in model accuracy at inference-time, yielding opportunities to run transformers more efficiently over applications. While quantization during training had previously shown promise (down to three bits, for most weights of the transformer), we observed the same reduction potential on attention values at application-time, allowing their representation to be reduced down to three bits (or even two for sentiment) with little effort (e.g., without retraining or using a dynamic quantization range). This shows it is feasible to implement efficient transformers by leveraging heavily sparse and quantized attention values, suggesting the possibility of building specialized hardware (e.g., FPGA and ASIC accelerators) to optimize the transformer's evaluation on-the-fly.

Acknowledgments

We would like to express our appreciation to Adithya V. Ganesan who assisted with our experiments.

This material is based upon work supported by the National Science Foundation under Grant Nos. 2007362 and 1918225. The experiments were conducted with equipment purchased through NSF Grant No. OAC-1919752.

References

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding Long and Structured Inputs in Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online. Association for Computational Linguistics.

J. Albericio, P. Judd, T. Hetherington, T. Aamodt, N. E. Jerger, and A. Moshovos. 2016. Cnvlutin: Ineffectual-Neuron-Free Deep Neural Network Computing. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pages 1–13. ISSN: 1063-6897.

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2020. BinaryBERT: Pushing the Limit of BERT Quantization. *arXiv:2012.15701 [cs]*. ArXiv: 2012.15701.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv:2004.05150 [cs]*. ArXiv: 2004.05150.

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. *arXiv:1906.00532 [cs]*. ArXiv: 1906.00532.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On Identifiability in Transformers. In *International Conference on Learning Representations*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The Lottery Ticket Hypothesis for Pre-trained BERT Networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 15834–15846. Curran Associates, Inc.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating Long Sequences with Sparse Transformers. *arXiv:1904.10509 [cs, stat]*. ArXiv: 1904.10509 version: 1.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively Sparse Transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2174–2184, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. 2019. Axial Attention in Multidimensional Transformers. *arXiv:1912.12180 [cs]*. ArXiv: 1912.12180.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Online. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. I-BERT: Integer-only BERT Quantization. *arXiv:2101.01321 [cs]*. ArXiv: 2101.01321.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the Dark Secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Lala Li and William Chan. 2019. Big bidirectional insertion representations for documents. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 194–198, Hong Kong. Association for Computational Linguistics.

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. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? In *Advances in Neural Information Processing Systems*, volume 32, pages 14014–14024. Curran Associates, Inc.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image Transformer. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4055–4064, Stockholmsmässan, Stockholm Sweden. PMLR.

Damian Pascual, Gino Brunner, and Roger Wattenhofer. 2021. Telling BERT's full story: from Local Attention to Global Aggregation. *arXiv:2004.05916 [cs]*. ArXiv: 2004.05916.

Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. 2020. Fully Quantized Transformer for Machine Translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*,

pages 1–14, Online. Association for Computational Linguistics.

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.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2020. Hopfield Networks is All You Need. *arXiv:2008.02217 [cs, stat]*. ArXiv: 2008.02217.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8(0):842–866. Number: 0.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. Poor Man's BERT: Smaller and Faster Transformer Models. *arXiv:2004.03844 [cs]*. ArXiv: 2004.03844.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked Language Model Scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8815–8821. Number: 05.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient Transformers: A Survey. *arXiv:2009.06732 [cs]*. ArXiv: 2009.06732 version: 1.

Jesse Vig. 2019. A Multiscale Visualization of Attention in the Transformer Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the Structure of Attention in a Transformer Language Model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2020. Similarity Analysis of Contextual Word Representation Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655, Online. Association for Computational Linguistics.

A. H. Zadeh, I. Edo, O. M. Awad, and A. 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.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8BERT: Quantized 8Bit BERT. *arXiv:1910.06188 [cs]*. ArXiv: 1910.06188.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020. Ternary-BERT: Distillation-aware Ultra-low Bit BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 509–521, Online. Association for Computational Linguistics.

A Consistency of Inducing Sparsity in the Attention

Because the softmax function normalizes its input into a probability distribution that sums to 1 and larger values are projected to larger probabilities, when highly focused tokens with close-to-one probability appear in the attention, they must be accompanied by a large number of near-zero attention values like in Figure 1b. Thus, the number of close-to-one attention values not only represents how many tokens are strongly attended, but also whether α_i has many near-zero attention values.

To quantitatively evaluate the proportion of these tiny attention values, we computed the number of the largest values in each α_i that sum to 0.5, visualizing their mean and standard deviation in Figure 4. On both pretrained RoBERTa and SQuAD-fine-tuned RoBERTa, we observed that most of the heads require on average fewer than ten attention values to sum up to 0.5, meaning that most heads focus strongly on fewer than ten tokens on average, leading to notable sparsity. We observe that seven of twelve heads in the first layers of both models have a larger average number (> 10) of such major tokens. For deeper layers, the average number of major tokens decreases. Finally, in the last two layers, we again see an increasing trend in the average number of major tokens. This indicates that middle layers commonly focus on only a small number of tokens, making these layers rich in sparsity. This confirms the "sparse deeper layers" identified by Correia et al. (2019); Clark et al. (2019) and further proves the existence of heavily focused tokens. It implies the large potential of inducing sparsity in the transformers and motivates us to explore how these sparse attention values contribute to the model accuracy. We also examined the BERT pretrained model and SQuAD-fine-tuned model, and we found behavior similar to RoBERTa. Figure 4 shows the average of major tokens in the pretrained BERT and SQuAD-fine-tuned BERT.

B Dispersion of Attention Histograms

Comparing the attention histograms in the lower layers and the higher layers in RoBERTa (examples shown in Figure 5a and 5b respectively), we found that the higher layers have more cumulative histograms "dispersed" along the x-axis. Together with the increasing variance of the number of major tokens in the last two layers shown in Figure 4, such a distribution pattern evidently expresses the greatly dissimilar sparsity among all the α_i in the head. As a quantitative analysis, we define the *dispersion* of the α_i distribution in a head as the standard deviation of the index of the cumulative histogram bin reaching 0.5. The dispersion expresses the dissimilarity of the α_i histogram. Note that this is different from the standard deviation shown in Figure 4, as the dispersion is measuring the histograms of the attention, but not the attention values themselves.

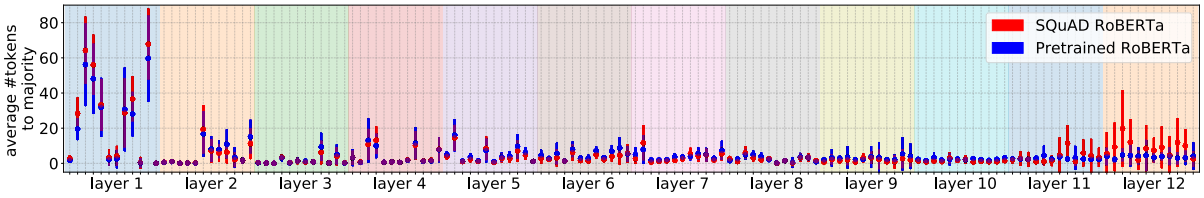
We measure the dispersion at each head along the layers for both pretrained and fine-tuned RoBERTa models. Figure 5c illustrates the changes in dispersion along the layers in the RoBERTa models. In pretrained RoBERTa and its SQuAD-fine-tuned version, the deep layers generally have higher dispersion. The difference between these two models is mainly in layer 11, where the pretrained model has a dispersion drop. RoBERTa fine-tuned for SST-2 does not show this trend. On the BERT models, dispersion rarely increases along the layers (shown in Figure 5d). The last layers have been proved to be task-specific (Wu et al., 2020; Rogers et al., 2020), and their attention can largely change after fine-tuning (Kovaleva et al., 2019). This potentially explains why we observed different dispersion behavior on different tasks, but needs further investigation.

C Heads with Outlier Attention Distribution

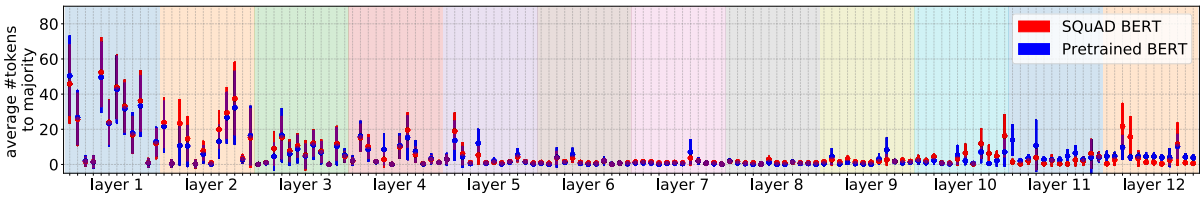
On some heads, a small portion of the tokens forms an attention histogram cluster separate from the majority, clearly showing a dissimilarity between these two types of distributions. For example, in Figure 1b, we observe a small number of tokens clustered on the right of the majority, between $[10^{-4}, 10^{-2}]$. Here we list all the heads with such pattern:

- Pretrained RoBERTa: Layer 1: head 8, head 10, head 12; Layer 2: head 3, head 5, head 10; Layer 3: head 2, head 10; Layer 4: head 4, head 9; Layer 5: head 2, head 7, head 10; Layer 6: head 5, head 11, head 12; Layer 7: head 3; Layer 8: head 7
- Pretrained BERT: Layer 3: head 10; Layer 5: head 5

We found that on these heads, the functional words/tokens and punctuation exhibit distributions that are significantly different from other tokens. For example, tokens such as `<s>`, `</s>`, `and`,

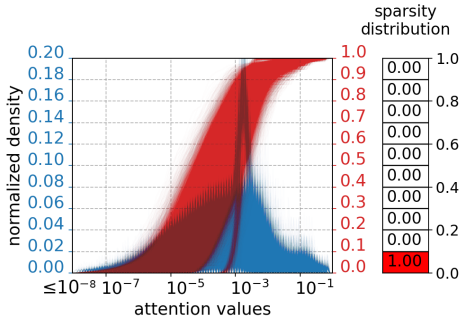


(a) RoBERTa

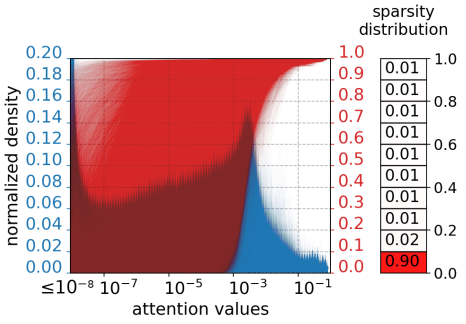


(b) BERT

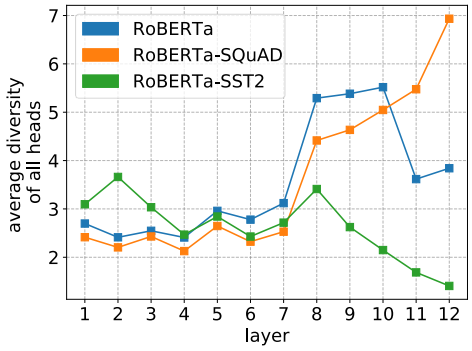
Figure 4: Mean and standard deviation of the number of tokens' attentions needed to cover a majority (i.e. sum to 0.5) of attention densities in both pretrained and SQuAD-fine-tuned RoBERTa/BERT models. Different layers are distinguished by different colors. In each layer the error bar represents the mean and std of head 1, head 2, ... , head 12 from the left to the right respectively.



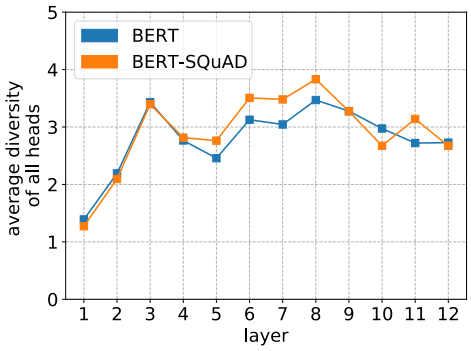
(a) Layer 1 head 1



(b) Layer 12 head 1



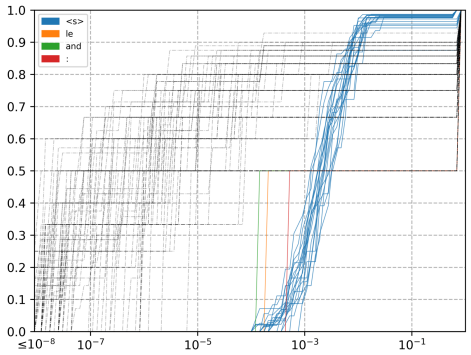
(c) Average dispersion of attention per layer in RoBERTa



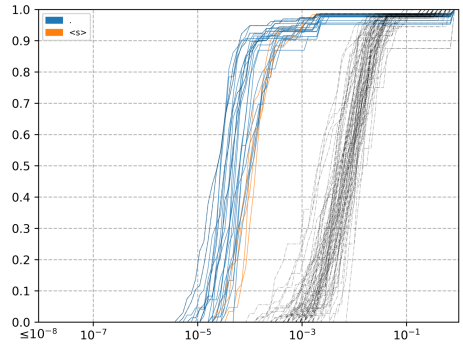
(d) Average dispersion of attention per layer in BERT

Figure 5: Attention distribution dispersion in different layers. Pretrained RoBERTa has more spread attention distributions in layer 12 than in layer 1. In (c), the pretrained and SQuAD-fine-tuned RoBERTa models exhibit increasing dispersion in deeper layers, while RoBERTa fine-tuned for SST-2 does not show such a trend.

: and . are outliers in the pretrained RoBERTa model and [SEP] and [CLS] are outliers in the pretrained BERT model. We also noticed these tokens' attention histograms could gather together like the majority of the tokens do, to form either a less sparse histogram cluster or more sparse histogram cluster, implying that on some heads, the functional words/tokens must be treated differently

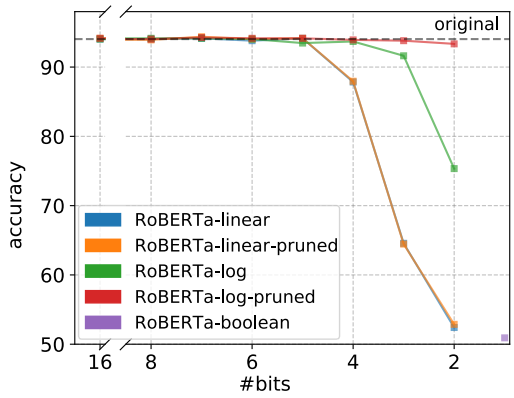


(a) Layer 2 head 3, <s>, le, and, : and </s> form a weak, less sparse cluster.

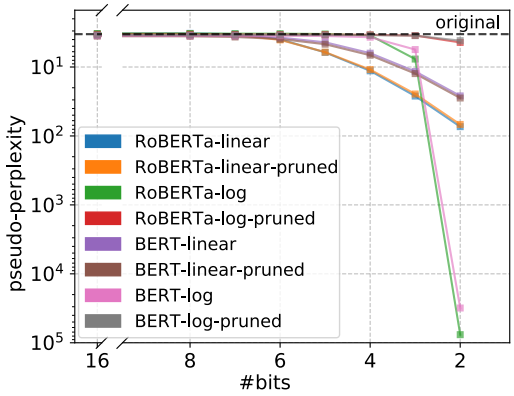


(b) Layer 4 head 4, <s> and . form a weak, more sparse cluster

Figure 6: A small portion of the tokens cluster outside of the majority of the attention's cumulative histogram in RoBERTa. Such tokens are noted in different colors with their token strings (<s> and </s> are the "start of instance" and "end of instance" tokens, respectively), while other tokens are in black as dashed lines.



(a) sentiment analysis



(b) masked language modeling

Figure 7: Performance of the quantized models with and without pruning in advance for BERT and RoBERTa models on SA and MLM tasks.

from the other tokens when exploring efficiency by utilizing sparsity. In Figure 6, we illustrate the attention histogram of such tokens. Our observation confirms that the special tokens and punctuation can be heavily attended (Voita et al., 2018; Clark et al., 2019; Kovaleva et al., 2019; Rogers et al., 2020). As a complement, we observed that it does not necessarily mean that the special tokens' attention are always more sparse than other tokens' attention.

D Quantization with Pruned Attention for SA and MLM

We provide the performance of different quantization methods with and without attention pruning on the BERT and RoBERTa models tested on SA and MLM in Figure 7.

E Quantization Methods and Their Effectiveness

Quantization methods. In Section 3, we implemented two different quantization methods. Algorithms 1 and 2 list their pseudo code.

Quantization and attention distribution Bhandare et al. (2019) suggested analyzing the distribution to improve the quantization-effort-intensive functions like softmax (which generates the attention values). Based on this, we assume that the transformer model will perform better if its quantized attention values are distributed similarly to the unquantized distribution. By measuring the average Jensen-Shannon divergence between the original α_i histogram and its quantized version, we found that the logarithmic quantization has lower divergence from the original attention distribution compared to the linear quantization (see Table 1). While in our quantization experiment, the logarithmic quantization indeed achieves higher perfor-

Algorithm 1: Linear quantization

input : *att*←attention values;
k←number of bits used for quantization;
t←pruning threshold
output : *res*←quantized attention values
quantile_size = $(1-t)/2^k$;
set quantized_value as middle point of quantile: quantile_size/2;
res=floor(*att* / quantile_size) * quantile_size + quantized_value + *t*;
set attention values less than quantile_size+*t* as zeros;

Algorithm 2: Log quantization

input : *att*←attention values;
k←number of bits used for quantization;
t←pruning threshold
output : *res*←quantized attention values
when not pruning *att*, choosing a small value 10^{-10} for *t*;
if *pruning att* **then**
  quantile_size = $(0-\log(t))/(2^k-1)$;
else
  quantile_size = $(0-\log(t))/(2^k)$
set quantized_value as middle point of quantile: quantile_size/2;
compute exponent of *res*: *exp_res*=floor((log(*att*) − log(*t*))/quantile_size)*quantile_size+quantized_value+*t*;
res=power(2, *exp_res*);
set values less than the first quantile boundry in the *res* as zeros;

mance than the linear quantization on most numbers of bits. This result indicates that selecting the quantization method with less divergence from the original attention distribution could improve the performance. However, the lower divergence between the quantized and original attention distribution does not necessarily relate to the model performance once we introduce pruning. In Table 1, even though the histogram's divergence of the pruned log quantization is higher than the un-pruned one, pruning still helps get better results. We hypothesize that the pruning enlarged the dissimilarity between the attention histograms, but such a change did not affect the accuracy since it only happened to the near-zero attention values.

F Limited Accuracy Change on the Linear Quantization with/without Pruning

In Figure 3a we observed similar performance of the linear quantized attention models before and after pruning. It is worth noting that the pruning threshold we selected, $\alpha < 10^{-3}$, is already a tiny value on the linear scale with respect to the range of the attention values $[0, 1]$. As a result, pruning will not significantly narrow the quantization range, as it does for the log-scale quantization. Thus the linear quantization has nearly the same effective quantized range with or without pruning, making it nearly impossible for the pruned linear quantized model to outperform the un-pruned one. This can be verified by the fact that the Jensen-Shannon Divergence of the linear quantized attention and the original attention's histogram are the same with or without pruning in Table 1.

quantization method	pruned	un-pruned
linear	0.67	0.67
log	0.58	0.55

Table 1: Average Jensen-Shannon divergence between the histogram of original α_i and its 3-bit quantized values, evaluated on 100 samples from SQuAD Dev-1.1. Log quantization, which has lower divergence from the original attention distribution, retains more accuracy from the original model.

G Experiment reproducibility

All evaluation is done on a server with the following specifications:

- CPU: Intel(R) Xeon(R) Silver 4216, 64 cores
- GPU: Quadro RTX 8000
- RAM: 377GB

The average runtime of the model inferences through the entire dataset is ∼4 hours, for different tasks. All datasets used in our experiment are based on English. The SQuAD tests are evaluated on 10570 sentences from the SQuAD Dev-v1.1 dataset. The SST2 tests are evaluated on 872 instances from the GLUE validation dataset. The Masked Language Modeling tests are evaluated on 480 paragraphs from the wikipedia training set, each having one random, unrepeated token masked for 15–25 iterations.