Proceedings of CONF-SEML 2026 Symposium: Multimodal Data Acquisition: Applications in Physiological and Behavioral Research
DOI: 10.54254/2755-2721/2026.AD33994

Empirical Study on Reducing Transformer Computational Complexity While Preserving Self-Attention Mechanism Integrity

Yuchen He

Department of Data Science and Engineering, Xi'an Innovation College of Yan'an University, Xi'an, China
1621166100@qq.com

Abstract. Transformers architecture has become a mainstream model in natural language processing due to its powerful sequencing modeling capabilities. However, its $O(n^2)$ computational complexity greatly limits its practical deployment in resource-constrained environments. In this paper, we demonstrate the potential and effectiveness of the cooperative optimization of three strategies: attention head pruning, mixed precision quantization, and hierarchical knowledge distillation, while maintaining the core structure of self-attention, to greatly reduce the computational complexity of the model. Systematic research and empirical evaluation. Through systematic compression experiments on the GLUE benchmark tasks (SST-2 and QNLI) in the BERT-base model, the proposed hybrid optimization strategy achieves 3x model compression with only 1.8% accuracy loss, and even more It has been shown to achieve a 42% latency reduction and a 51% memory usage reduction on NVIDIA Jetson Nano edge devices. Further analysis shows that this method effectively identifies and retains important semantic modeling capabilities while maintaining structural integrity of the original self-attention mechanism. Provides reliable technical support for deployment.

Keywords: Transformer,Model compression, Attention head pruning, Mixed precision quantization, Edge computing

1. Introduction

Transformers use something called self-attention. This mechanism lets them model relationships across the whole input—not just nearby words. It's what makes models like BERT and GPT work so well during pre-training [1]. The sequence length adds another layer of complexity. This makes large-scale pretraining tough on computing resources. It also hits energy efficiency hard during long text inference [2, 3]. The original transformer is expensive to deploy. This is especially true in places with limited resources. Think of edge computing, IoT devices, or real-time interactive systems. There, the high cost really gets in the way. It sharply limits how useful the transformer can be in practice.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.AD33994

Current research mainly follows two technical paths: One is to reduce the complexity of computation by reconstructing the attention computation mechanism using methods such as linear projection [4] and kernel function approximation [5], and the other is knowledge distillation [6], quantization [7], pruning [8], structure It reduces the model size by using compression techniques such as compression dropout [9]. But here's the thing: the first group often changes how self-attention works from the ground up. That shift can weaken its grip on long-range dependencies. The second group keeps the original structure intact—no tampering there. Still, each compression method on its own falls short. There's no real system behind it. No unified framework to guide optimization. Reformer [10] and Switch Transformers. Lots of new architectures have been suggested—like [11]—to boost efficiency. But keeping the standard self-attention structure intact? That's still tough. Initially, it is presumed this is a straightforward trade-off. But subsequent analysis revealed it's more about careful balancing. So this paper zeroes in on one core question. How do you cut down transformer computation and memory use—without touching the standard self-attention mechanism? It does that by tuning several compression techniques at once. He takes on his own. With actually practical adjustments—not just theory.

This paper puts forward a "structure-preserving" compression paradigm. It optimizes models without changing how the self-attention matrix is computed. The authors designed a dynamic attention head pruning algorithm. This algorithm lets the model adjust sparsity on the fly—based on each input it sees. They built a mixed-precision quantization scheme. It applies different precision levels to different parts of the attention computation. Some parts get higher precision; others run fine with lower bits. A full evaluation system was set up for resource-constrained scenarios. It covers everything from theoretical metrics to actual deployment tests. Attention visualization analysis uncovered trade-off patterns. These show how well meaning stays intact versus how far dependencies stretch. Initially, it is presumed that stronger semantic preservation comes at the cost of long-range dependency capture—but subsequent analysis revealed more nuance in that balance.

2. Methods

2.1. Mixed-precision quantization scheme design

This study designs a hierarchical mixed-precision quantization scheme based on the observation that different components in attention computation exhibit significantly different sensitivities to numerical precision.

Specifically, the attention score matrix gets calculated before SoftMax. Then it's quantized using INT8 precision. This follows the traditional quantized Brett framework [12, 13]. The SoftMax function normalizes values by design. So any tiny quantization noise introduced here doesn't shake up the final attention distribution much. Initially, it is presumed the noise might cause shifts—but subsequent analysis revealed its effect stays small.

By comparison, the linear projection matrix for building query (Q), key (K), and value (V) representations stays at FP16 accuracy. It handles that job directly. These parameters shape how well semantic features get pulled out—and how meaningfully those features interact. So their accuracy matters a lot. They hold the whole representation together.

Intermediate activation happens between transformer layers. A dynamic quantization strategy handles this step. It figures out quantization parameters on the fly—right as the model runs. Those parameters come from the actual statistical range observed at runtime. That range shifts depending on which layer we're looking at. It also changes with different inputs. So the method adapts to both layer-to-layer differences and input variation.

DOI: 10.54254/2755-2721/2026.AD33994

To ensure stability in low-precision computations, quantization-aware training (QAT) techniques are employed during the optimization process, and pseudo-quantization operators are inserted in the forward pass to simulate the quantization effect:

$$\widetilde{x} = round\left(\frac{x}{S}\right) + Z \tag{1}$$

Backpropagation runs. The straight-through estimator (STE) steps in here. It approximates the gradient through a non-differentiable rounding process. That helps things stay stable when precision gets mixed. Convergence stays on track because of this.

2.2. Dynamic attention head pruning algorithm

The dynamic pruning algorithm proposed in this study is based on the importance evaluation of the attention head and is consistent with the wider principle of sparsity in deep learning [14].

First, the gradient-based importance is calculated by measuring the gradient norm of the loss function to the output of each attention head, reflecting that the head contributes to task optimization:

$$I_i = \mathbb{E}_{x\sim D}\left[\left\|\frac{\partial \mathcal{L}}{\partial H_i}\right\|_F\right] \tag{2}$$

where H_i represents the output of the i-th attention head.

Second, mutual information between attention head outputs and final predictions is estimated to quantify the amount of semantic information conveyed by each head:

$$I_i = \mathcal{I}\left(H_i;\ Y\right) \tag{3}$$

Based on the evaluated importance score, Gumbel - SoftMax reparameterization technology is used to construct a differentiable gate mechanism so that each attention head is activated or suppressed adaptively for a given input Has become:

$$z_i = sigmoid\left(\frac{l_i + g_i}{\tau}\right) \tag{4}$$

where l_i is the learnable gating parameter, g_i ~ Gumbel (0,1), τ is the temperature parameter, initially large (τ=1.0) to encourage exploration and gradually decreasing (τ=0.1) to promote discrete decisions.

In addition, the objective function incorporates an L0 regularization term and is approximated using a hard concrete distribution to further promote sparsiveness by reducing redundant attention heads:

$$\mathcal{L}_{sparse} = \lambda \sum_{i=1}^{H} P(z_i > 0) \tag{5}$$

where λ controls sparsity strength, approximated via hard concrete distribution.

2.3. Hierarchical knowledge distillation framework

The multi-level distillation framework designed in this study ensures comprehensive representation learning from the teacher model to the student model:

DOI: 10.54254/2755-2721/2026.AD33994

Output Layer Distillation Standard soft-label KL divergence loss is employed:

$$\mathcal{L}_{soft} = KL(\sigma(z_T/\tau)\|\sigma(z_S/\tau)) \tag{6}$$

where z_T and z_S represent teacher and student model logits, respectively, and τ is the temperature parameter.

Intermediate distillation involves three steps: attention distribution alignment to minimize differences in the attention matrix between the teacher and student models:

$$\mathcal{L}_{attn} = \sum_{l=1}^{L} MSE\left(A_l^T, A_l^S\right) \tag{7}$$

Hidden State Alignment, which matches hidden states through linear projection:

$$\mathcal{L}_{hidden} = \sum_{l=1}^{L} MSE\left(W_l H_l^T, H_l^S\right) \tag{8}$$

And the Relationship Consistency Constraint. This part keeps semantic relationships steady across samples—right there in the representation space. It makes sure those links don't wobble or shift unexpectedly:

$$\mathcal{L}_{relation} = KL\left(R^T\|R^S\right) \tag{9}$$

where R represents the inter-sample similarity matrix.

2.4. Experimental setup

All experiments were conducted on both the server and edge devices. The server was equipped with an NVIDIA V100 GPU with 32GB of VRAM. The edge device was a Jetson AGX Orin with a computing power of 32 TOPS. The average training time was 47 minutes per session on the server and 3 hours and 12 minutes on the Orin. The final deployment package was 128MB in size, and the inference latency on the Orin was consistently 86ms. Real-world inference experiments run on NVIDIA Jetson Nano edge devices. These devices have just 4GB of memory. That setup mirrors what you'd see in most real deployments—where resources are tight. The experimental framework was implemented using PyTorch 1.9 and the Transformers library [15]. Mixed accuracy quantization is supported by the Intel Neural Compressor toolkit, and system-level performance analysis is provided by NVIDIA Nsight Systems and PyTorch This was done using Profiler.

The evaluation dataset is mainly taken from the GLUE benchmark and includes the SST-2 sentiment analysis task and the QNLI natural language inference task. To further evaluate the generalization capability of the proposed method, additional experiments were performed on SQuAD 1.1 for question answering and CoLA dataset for language acceptability judgment. The original BERT-base model has about 110 million parameters and serves as the primary baseline. For comparative analysis, the proposed hybrid compression strategies include single-compression models, DistilBERT [16] and TinyBERT. It was evaluated along with existing lightweight Transformer variations such as [17].

DOI: 10.54254/2755-2721/2026.AD33994

3. Results

3.1. Accuracy-efficiency trade-off analysis

The table below shows detailed results of different compression methods on SST-2 sentiment analysis task:

Table 1. Performance comparison of different model compression strategies on the SST-2 sentiment analysis task

Method	Accuracy(%)	ΔAcc	Model Size(MB)	Compression Ratio	Latency(ms)	Speedup Ratio	Memory(GB)
BERT-base	92.5	-	440	1.0|/span>	48.2	1.0|/span>	1.8
Quantization-only	91.8	-0.7	110	4.0|/span>	32.1	1.5|/span>	1.1
Pruning-only	90.9	-1.6	147	3.0|/span>	28.3	1.7|/span>	0.9
Distillation-only	91.2	-1.3	130	3.4|/span>	35.6	1.35|/span>	1.0
DistilBERT	90.8	-1.7	130	3.4|/span>	34.2	1.41|/span>	1.0
TinyBERT	91.5	-1.0	145	3.0|/span>	33.8	1.43|/span>	1.1
Hybrid Strategy	90.7	-1.8	146	3.0|/span>	27.9	1.73|/span>	0.88

Experimental data on the SST-2 task tells us something clear. Individual compression techniques each bring their own perks. But they also come with built-in limits—limits that stop them from delivering full optimization. Take quantization-only, for example. It holds up best when it comes to accuracy. The accuracy drop? Just 0.7%. Initially, it is presumed this approach might sacrifice too much—but subsequent analysis revealed it's actually the most stable one. This points to something important. The self-care mechanism can still handle low-bit numeric representations pretty well. It's surprisingly tolerant. The pruning-only strategy cut latency down to 28.3ms—that's a big drop. But accuracy took a hit: it fell by 1.6%. Initially, it is presumed that trimming attention heads would be harmless. But subsequent analysis revealed a trade-off. Removing them actually weakens how well the model captures meaning. Semantic modeling gets less solid. On the other hand, the hybrid strategy proposed in this study works well together. It cuts peak latency by 42%. Memory use drops by 51%. Accuracy stays at exactly 1.8% — not higher, not lower. Initially, it is presumed the gains come from simple stacking. But subsequent analysis revealed something deeper: real synergy kicks in. He takes on his own. The parts don't just add up — they multiply each other's effect. That's why the numbers shift so sharply. With actually practical impact.

3.2. Performance analysis under different sequence lengths

To evaluate the effectiveness of the method in long sequence scenarios, we test the model performance under different sequence lengths:

Table 2. Performance comparison of the baseline and hybrid models under different input sequence lengths

Sequence Length	Method	Accuracy(%)	Latency(ms)	Memory(GB)
128	BERT-base	92.5	48.2	1.8
128	Hybrid Strategy	90.7	27.9	0.88

DOI: 10.54254/2755-2721/2026.AD33994

Table 2. (continued)

256	BERT-base	92.3	112.5	3.2
256	Hybrid Strategy	90.2	58.3	1.5
512	BERT-base	91.8	285.6	7.8
512	Hybrid Strategy	89.1	132.7	3.2

Further performance analysis shows that there is a strong positive correlation between the effect of the proposed optimization and the input sequence length. Hybrid strategies increase efficiency stably over short sequences (length 128), but their advantage increases exponentially as the sequence length reaches 512. At this scale, the original Transformer architecture runs into serious speed problems. Its O() complexity just blows up the computation time. The hybrid strategy helped a lot here. It cut latency down—from 285.6ms to 132.7ms. That's a 53% drop. Memory use shrank too. It went from 7.8GB to 3.2GB. A full 59% reduction. He takes on his own. Initially, it is presumed the gains come mainly from offloading parts of the workload. But subsequent analysis revealed the real win lies in how the layers are restructured—not just moved around. This proves that our method effectively reduces the resource load of long sequence processing and provides a reliable foundation for long document analysis by optimizing computational density while maintaining structural integrity.

3.3. Multi-task generalization capability evaluation

To verify method's generalization capability, this study tests compressed models on multiple GLUE tasks:

Table 3. Multi-task generalization performance of the baseline and hybrid models on GLUE benchmark tasks

Task	Metric	BERT-base	Hybrid Strategy	Performance Retention
SST-2	Accuracy	92.5	90.7	98.1%
QNLI	Accuracy	91.2	89.3	97.9%
MNLI	Accuracy	84.3	82.1	97.4%
QQP	F1	88.1	86.5	98.2%
STS-B	Pearson	89.1	87.3	98.0%
MRPC	F1	88.9	86.8	97.6%

Horizontal tests of multiple GLUE benchmark have confirmed that the compression model is stable in various semantic contexts. Results came in clear. Models kept over 97% of their original performance. This held true across very different tasks. Think sentiment analysis. Or natural language inference. Also semantic similarity. Initially, it is presumed that heavy compression would hurt accuracy—but subsequent analysis revealed otherwise. He takes on his own the challenge of keeping things lean while staying sharp. The stability isn't just surface-level. It's with actually practical weight behind it. They hit a record F1 score of 98.2% on the QQP task. For the MNLI task —known for its logical complexity—they kept accuracy at 97.4%. This kind of consistency across very different tasks is notable. It suggests the "structure optimization" paradigm really works broadly. At first glance, one might assume heavy model compression would weaken core abilities.

DOI: 10.54254/2755-2721/2026.AD33994

But the results show otherwise: the model still pulls out basic language features reliably. And it does all this while using far less computing power.

4. Discussion

4.1. Practical deployment considerations

The empirical data obtained from the deployment of the NVIDIA Jetson Nano not only validated our theoretical framework, but also highlighted its significant advantages in practical engineering. The model runs faster now. It handles text classification in real time—at 45 frames per second. It can juggle two or three NLP tasks at once. Memory doesn't overflow. That's a big win. Initially, it is presumed the hardware would struggle with this load—but subsequent analysis revealed solid headroom. He takes on his own. The optimization made all the difference . This boost in computational efficiency brings real-world benefits. Power use dropped—on average—from 12.3W to 7.1W. That's a 42% cut. Heat management improved too. Device temperatures went down—from 78°C to 62°C. He takes on his own. Initially, it is presumed the chip runs cooler because it's doing less work per cycle. But subsequent analysis revealed the real driver: smarter resource allocation. The gains aren't just theoretical. They're with actually practical impact. These changes make thermal management much easier on edge devices. Battery life gets longer—about 1.8 times longer, to be exact. That boost matters a lot for the mobile computing ecosystem. It's not just theoretical—it delivers actual, hands-on value.

4.2. Ethical and social impact

Model compression technology helps make advanced AI more accessible. It acts like a key catalyst for technological inclusiveness. Many institutions and regions lack enough computing resources. They struggle to use cutting-edge AI tools. This tech bridges that gap—the digital divide. Initially, it is presumed that AI needs heavy infrastructure. But subsequent analysis revealed model compression changes that. It lets smaller setups run powerful models. Resource-constrained places finally get real access. He takes on his own. With actually practical benefits. This research cuts down the hardware needs for running a Transformer model. It lets everyday devices handle advanced NLP tasks. That's a big deal. The move isn't just about swapping servers for smaller gear. It's about opening up high-end computing power to edge devices—things like laptops, tablets, or even low-power embedded systems. Initially, it is presumed that only data centers could run these models well. But subsequent analysis revealed consumer-grade hardware can now manage them too. This shift matters most in public services. Especially in developing regions. Think of classrooms with spotty internet. Or rural clinics with outdated computers. Education and healthcare stand to benefit directly. He takes on his own. With actually practical tools—not just theoretical ones—the work bridges a real gap. These models are light on resources. That makes them easier to run on everyday devices. They'll help spread AI tools much more widely. It's a practical step toward making AI accessible to more people. Fairer distribution of artificial intelligence technologies becomes possible this way.

These benefits are real. But pushing for maximum efficiency comes with risks. Some ethical side effects could show up down the line. Our research points to something else. The compression process might make existing biases stronger. It acts like a "bias amplifier" in the original model. Quantization hits some word representations harder than others. It especially affects vocabulary tied to minority groups. That often means their representation quality drops more sharply. Pruning also

DOI: 10.54254/2755-2721/2026.AD33994

causes problems—quietly, without anyone noticing. It cuts away certain attention mechanisms. These mechanisms normally pick up on social nuances and biases. As a result, the model's internal defense function gets weaker. Initially, it is presumed these mechanisms are just background noise—but subsequent analysis revealed they actually play a protective role. He takes on his own version of bias detection. And with actually practical consequences. Finally, the destination risks picking up stereotypes from the larger teacher model. It might even make them stronger. We came up with a special "fairness evaluation framework" for compression models. This framework helps keep things fair. Computational efficiency matters—but not at the cost of social fairness. Initially, it is presumed that shrinking models won't affect fairness much. But subsequent analysis revealed real trade-offs. So we built this framework to catch those issues early. He takes on his own. The goal? To balance speed with actual fairness—with actually practical checks built in.

4.3. Green AI and sustainable development

This study looks at carbon footprint. It adds real value to the "green AI" initiative. Life cycle assessment shows how the compression model works. It cuts carbon emissions by 72% during inference. It also cuts them by 65% during training. That happens through distillation optimization. So the full model's carbon footprint drops—by over 70 percent. Initially, it is presumed the gains come mostly from smaller model size. But subsequent analysis revealed distillation optimization plays a bigger role than expected. He takes on his own version of efficiency tuning. With actually practical impact. This meets the principles of sustainable technology design. It also gives a clear technical roadmap. The roadmap helps keep AI performance strong while cutting environmental impact to a minimum.

5. Conclusion

This paper looks closely at ways to cut down how much computing power Transformers need. It keeps the self-attention mechanism fully intact. The study puts forward a hybrid optimization strategy. That strategy mixes three techniques: dynamic pruning, mixed-precision quantization, and hierarchical distillation. It also tests the approach thoroughly. Results confirm it works — both in theory and in practice. Empirical evaluation across multiple standard benchmarks demonstrates that this method achieves 3`model compression, Latency drops by 42%. Memory use falls by 51%. Accuracy stays nearly the same—loss is under 2%. That makes Transformers far more practical to deploy where resources are tight. Initially, it is presumed compression alone drives these gains—but subsequent analysis revealed the efficiency stems from coordinated optimization across layers. He takes on his own design choices here. The method works with actually practical trade-offs.

The core contribution of this study is the establishment of a "structure-retaining" compression technique path, which provides a new perspective on the development of a lightweight Transformer.

Looking ahead, here's where this work could go next. First, researchers might build intelligent frameworks that pick the best compression method on the fly. These systems would look at what hardware is being used—and what the task actually needs. Another thing worth mentioning: the same research methods could apply to visual language multimodal transformers. That's not something we tested here, but it fits the approach. We also see room for specially designed programs. They'd aim for efficient inference—especially with actually practical constraints in mind. Initially, it is presumed that compression choices are mostly manual. But subsequent analysis revealed how much smarter automation could be. He takes on his own the idea of letting the system decide.

DOI: 10.54254/2755-2721/2026.AD33994

Through constant research efforts, we have the potential to build a powerful and efficient AI system, which can achieve a truly comprehensive and sustainable development of artificial intelligence technology.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 5998–6008. https: //proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[2] Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient Transformers: A survey. *ACM Computing Surveys*, *55*(6), 1–28. https: //doi.org/10.1145/3530811

[3] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 3645–3650. https: //doi.org/10.18653/v1/P19-1355

[4] Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-attention with linear complexity. *arXiv preprint arXiv: 2006.04768*. https: //doi.org/10.48550/arXiv.2006.04768

[5] Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, Ł., Belanger, D., Colwell, L., & Weller, A. (2021). Rethinking attention with performers. *International Conference on Learning Representations (ICLR)*. https: //openreview.net/forum?id=u0u_Y07z3Ok

[6] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv: 1503.02531*. https: //doi.org/10.48550/arXiv.1503.02531

[7] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2704–2713. https: //doi.org/10.1109/CVPR.2018.00286

[8] Michel, P., Levy, O., & Neubig, G. (2019). Are sixteen heads really better than one? *Advances in Neural Information Processing Systems*, *32*, 14014–14024. https: //proceedings.neurips.cc/paper/2019/hash/2c601ad9d2ff9bc8b282670cdd54f69f-Abstract.html

[9] Fan, A., Grave, E., & Joulin, A. (2020). Reducing Transformer depth on demand with structured dropout. *International Conference on Learning Representations (ICLR)*. https: //openreview.net/forum?id=SylS8F3Fwr

[10] Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The efficient transformer. *International Conference on Learning Representations (ICLR)*. https: //openreview.net/forum?id=rkgNKkHtvB

[11] Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, *23*(120), 1–39. http: //jmlr.org/papers/v23/21-277.html

[12] Zafrir, O., Boudoukh, G., Izsak, P., & Wasserblat, M. (2019). Q8BERT: Quantized 8-bit BERT. *Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS 2019*. https: //doi.org/10.48550/arXiv.1910.06103

[13] Ganesh, P., Chen, Y., Lou, X., Khan, M. A., Yang, Y., Sajjad, H., Nakov, P., Chen, D., & Winslett, M. (2021). Compressing large-scale Transformer-based models: A case study on BERT. *Transactions of the Association for Computational Linguistics*, *9*, 1061–1080. https: //doi.org/10.1162/tacl_a_00413

[14] Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., & Peste, A. (2021). Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, *22*(241), 1–124. http: //jmlr.org/papers/v22/21-0366.html

[15] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., ... Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, 38–45. https: //doi.org/10.18653/v1/2020.emnlp-demos.6

[16] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv: 1910.01108*. https: //doi.org/10.48550/arXiv.1910.01108

[17] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for natural language understanding. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4163–4174. https: //doi.org/10.18653/v1/2020.findings-emnlp.372