Journal of computer science and software applications
https://www.mfacademia.org
ISSN:2377-0430
Vol. 3, No. 4, 2025

Efficient Compression of Large Language Models with Distillation and Fine-Tuning

Anda Kai[1], Lin Zhu[2], Jiangchuan Gong[3]
[1]The University of Texas at Austin, Austin, USA
[2]Stevens Institute of Technology, New Jersey, USA
[3]Hebei Normal University, Shijiazhuang, China
*Corresponding Author: Anda Kai; anda.kai@utexas.edu

Abstract:

With the widespread adoption of large language models (LLMs), their extensive parameter scale and high computational cost pose significant challenges for practical deployment. To address this issue, this study proposes a method that integrates Knowledge Distillation and Parameter-Efficient Fine-Tuning (PEFT) to reduce computational overhead while preserving high performance. In the knowledge distillation phase, experiments are conducted using different temperature parameters to analyze their impact on student model learning. The role of various feature distillation levels in model compression is also explored. Experimental results indicate that moderate temperature parameters enhance the distillation effect. Moreover, selecting an appropriate feature layer for distillation improves the generalization ability of the student model. In the fine-tuning phase, the performance of LoRA (Low-Rank Adaptation) is compared with full fine-tuning. Results show that LoRA offers significant advantages in inference speed and computational efficiency, whereas full-parameter fine-tuning achieves superior accuracy and language understanding. Comprehensive experimental findings confirm that a well-designed combination of knowledge distillation and fine-tuning can achieve effective model compression while maintaining performance. Future research can integrate additional compression techniques, such as pruning and quantization, to further enhance model adaptability and computational efficiency. This approach provides a promising solution for deploying large-scale language models in low-resource environments.

Keywords:

Parameter-Efficient Fine-Tuning, Knowledge Distillation, Large Language Model, Model Optimization

1. Introduction

With the rapid advancement of deep learning, Large Language Models (LLMs) have demonstrated outstanding performance in Natural Language Processing (NLP) tasks [1]. However, these models require extensive computational resources due to their massive parameter scale, leading to high deployment costs. In edge computing, mobile devices, and low-resource environments, directly applying large-scale language models poses significant challenges [2]. Therefore, achieving lightweight optimization while maintaining model performance has become a crucial issue for both academia and industry. To address this, Knowledge Distillation (KD) and Fine-Tuning have emerged as key strategies to reduce model complexity and enhance computational efficiency while preserving the capabilities of large models. These methods provide an effective solution for deploying large language models [3].

Knowledge distillation is a model compression technique based on the teacher-student framework. It utilizes a pre-trained teacher model as a knowledge source, transferring learned features and decision patterns to a smaller student model. This enables model compression while retaining strong inference capabilities. Techniques such as Soft Labels, Intermediate Feature Matching, and Attention Distillation help reduce parameter count without significantly degrading performance. Additionally, fine-tuning plays a critical role in adapting models to specific tasks. Rather than training a small model from scratch, fine-tuning a large pre-trained model leverages existing knowledge, reduces training costs, and optimizes task-specific performance. The combination of knowledge distillation and fine-tuning has thus become a central approach in lightweight model research.

Recent studies have explored various lightweight strategies that integrate knowledge distillation and fine-tuning. Parameter-Efficient Fine-Tuning (PEFT) methods, such as LoRA (Low-Rank Adaptation) and Adapter-based techniques, allow models to adapt to new tasks with minimal additional parameters. These approaches enhance flexibility without substantially increasing computational overhead. Additionally, techniques like Pruning and Quantization further refine model structures, reducing redundant computations and improving efficiency. However, balancing model compression with performance retention remains a major challenge. For instance, in knowledge distillation, selecting an effective distillation strategy and designing an appropriate loss function are critical to ensuring that the student model inherits the capabilities of the teacher model without suffering performance degradation due to oversimplification [4].

In practical applications, lightweight LLMs combining knowledge distillation and fine-tuning have been widely explored across multiple domains. In intelligent voice assistants, text summarization, and mobile-based question-answering systems, these models offer near-original reasoning capabilities with significantly lower computational costs and latency. In enterprise applications such as finance and healthcare, lightweight models enhance inference speed while reducing server-side computational burdens, improving overall system throughput. Given the diverse requirements of different application scenarios, optimizing the combination of knowledge distillation and fine-tuning remains a key research focus [5].

In conclusion, research on LLM lightweight optimization is essential for facilitating their widespread adoption in real-world applications. The integration of knowledge distillation and fine-tuning provides a promising solution to the trade-off between model size and performance. As advancements in computing hardware and optimization techniques continue, further reducing computational resource consumption while maintaining model effectiveness will be crucial for the development of large language models. This study aims to explore lightweight optimization strategies based on knowledge distillation and fine-tuning, providing both theoretical insights and practical guidance for improving computational efficiency and reducing deployment costs.

2. Method

In this study, we propose a lightweight method for large language models that combines knowledge distillation with fine-tuning techniques, aiming to maintain the high performance of the models while reducing the computational cost [6]. The overall model architecture is shown in Figure 1.

Figure 1 illustrates the overall architecture of the proposed method. The teacher model, a large pre-trained language model, generates both logits and intermediate attention features, which are transferred to the student model through two forms of distillation: output-level knowledge distillation and hidden-layer

feature distillation. The student model learns from the teacher's soft labels and intermediate representations to enhance its performance while maintaining a lightweight structure. This integrated approach ensures that the student model not only mimics the final output of the teacher but also captures rich semantic information from intermediate layers, thereby improving generalization and compression effectiveness.

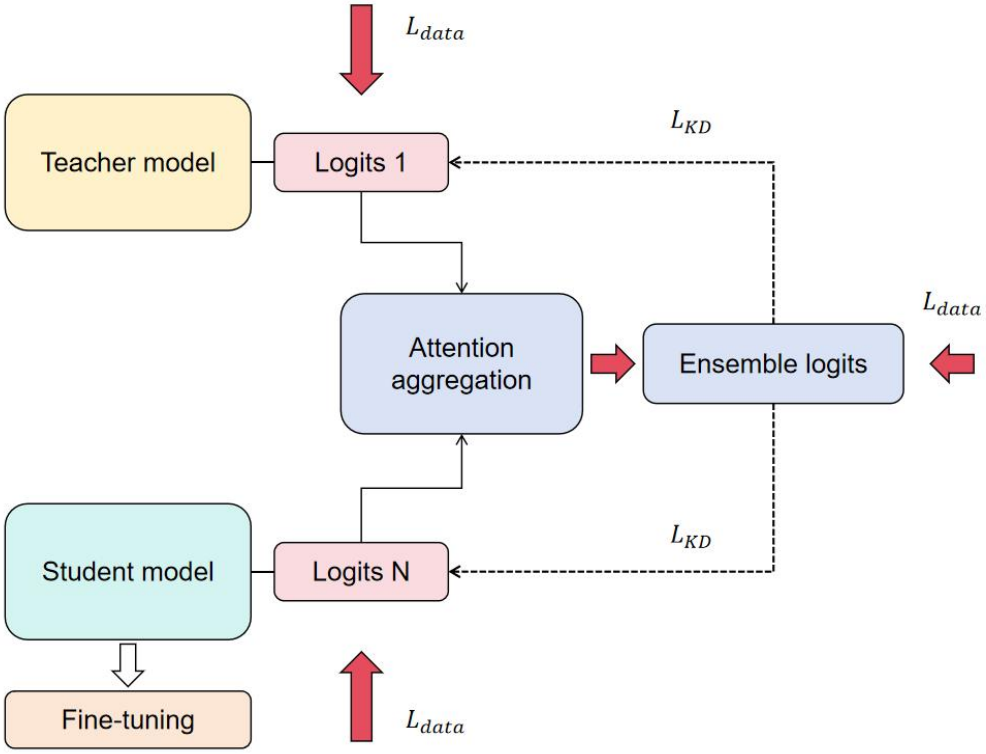

Figure 1. Overall model architecture

First, knowledge distillation adopts a teacher-student framework, where teacher model $T(x)$ is a pre-trained large language model and student model $S(x)$ is a smaller lightweight model. The traditional knowledge distillation loss function consists of two parts, one is the cross-entropy loss L_{CE} based on hard label, the other is the Kullback-Leibler (KL) divergence loss L_{KD} based on soft label.

$$L_{KD} = \tau^2 \sum_i p_i^T \log \frac{p_i^T}{p_i^S}$$

Where, p^T and p^S represent the output probability distribution of the teacher model and the student model respectively, and τ is the temperature coefficient. When the temperature coefficient τ increases, the probability distribution becomes smoother, which helps the student model to capture more implicit information. In this study, on the basis of the standard distillation loss, the intermediate layer feature distillation is introduced, that is, the Euclidean distance between the teacher model and the student model in the hidden layer is minimized:

$$L_{feature} = \sum_{l} \| h_l^T - h_l^S \|^2$$

Where, h_l^T and h_l^S represent the hidden state of layer l, respectively. In this study, the characteristics of different layers were selected and matched to improve the distillation effect.

In the Fine-tuning stage, this study adopts Parameter-Efficient fine-tuning (PEFT) methods, such as LoRA (Low-Rank Adaptation). LoRA adds a trainable low-rank matrix to the weight matrix of Transformer to reduce the amount of parameter updates. Its mathematical expression is as follows:

$$\Delta W = AB, \quad A \in R^{dxr}, B \in R^{rxd}, r << d$$

Where W is the weight of the original model, and A and B are low-rank matrices. In this study, the LoRA structure is optimized so that it can adapt to the lightweight student model and combined with knowledge distillation to form an end-to-end optimization strategy.

In addition, Pruning and Quantization techniques were combined to further reduce the size of the model. Pruning methods reduce computation by removing weights of low importance, while quantization maps model parameters from 32-bit floating-point numbers to representations of 8-bit or less precision, as follows:

$$w_q = round\left(\frac{w - w_{\min}}{w_{\max} - w_{\min}} \times (2^b - 1)\right)$$

Where, w_q is the weight after quantization, b is the number of bits, and $w_{\min}$ and $w_{\max}$ are the minimum and maximum values of the weight. In this study, different quantization strategies are tested on different tasks and their effects on model performance are analyzed.

Based on the above methods, an optimization framework combining knowledge distillation, fine-tuning, pruning and quantization is proposed to realize the lightweight of large language models. In the experimental part, the performance of different strategies will be compared, and the effects of lightweight on inference speed, memory occupancy and task accuracy will be analyzed to verify the effectiveness of the proposed method[7].

3. Experiment

3.1 Datasets

In this study, the OpenWebText dataset serves as the primary data source for training and distillation. OpenWebText is a high-quality open text corpus designed to replicate the WebText dataset used by OpenAI for training the GPT series models. It consists of highly rated articles from Reddit, covering diverse topics such as technology, finance, healthcare, and law. This broad coverage provides rich semantic information for both knowledge distillation and fine-tuning. Additionally, OpenWebText undergoes strict de-duplication and data cleaning processes to ensure text quality, eliminating low-quality content that could interfere with model learning. These characteristics make it well-suited for large language model training and lightweight optimization research.

In the knowledge distillation stage, a subset of OpenWebText is used to generate soft labels from the teacher model. The study randomly selects texts from multiple categories to enhance data diversity and improve the generalization ability of the student model. Soft labels are generated under different temperature settings to help the student model adapt to long-tail distributed data. OpenWebText is also used for fine-tuning. A small number of high-quality examples are employed to train the student model, improving its performance on specific tasks and enhancing the utility of lightweight models.

To assess the effectiveness of the lightweight approach, this study evaluates model performance on the OpenWebText test set. The impact of different knowledge distillation and fine-tuning strategies is analyzed in terms of compression rate, inference speed, and text generation quality. Additionally, to examine real-world applicability, the model is tested on common NLP tasks, such as text summarization and question-answering. These experiments provide a comprehensive evaluation of the proposed method and offer valuable insights for further optimization.

3.2 Experimental Results

First, this paper conducted a comparison experiment of knowledge distillation effect under different temperature parameters, and the experimental results are shown in Table 1.

Table 1: Comparative experiment of knowledge distillation effect under different temperature parameters

Temperature parameter	Model accuracy	Perplexity	Rate of inference (Tokens/s)
0.5	82.3	18.5	45
1.0	85.7	15.8	42
2.0	87.1	14.2	39
5.0	83.9	17.6	41
10.0	78.5	22.9	47

According to the experimental results, temperature parameters have a significant impact on knowledge distillation, particularly in terms of model accuracy and perplexity [8]. At a low temperature setting, the model achieves an accuracy of 82.3% and a perplexity of 18.5. This suggests that the student model relies heavily on the deterministic outputs of the teacher model and struggles to capture the probabilistic information embedded in the distribution. As the temperature increases to 1.0 and 2.0, accuracy improves to 85.7% and 87.1%, respectively, while perplexity decreases to 15.8 and 14.2. These results indicate that a moderate increase in temperature enables the student model to learn smoother and more comprehensive knowledge, enhancing its generalization ability.

However, when the temperature is further raised to 5.0 and 10.0, accuracy declines to 83.9% and 78.5%, while perplexity rises to 17.6 and 22.9. This suggests that excessively high temperatures cause the probability distribution to become overly smooth. As a result, the student model struggles to differentiate between categories, weakening the learning process. In such cases, the effectiveness of knowledge distillation diminishes, and the student model fails to acquire the teacher model's key decision-making capabilities. Additionally, in terms of inference speed, temperature variations have minimal impact, with

fluctuations between 39 and 47 tokens per second. This indicates that temperature primarily influences the model's learning effectiveness rather than its inference efficiency [9].

In summary, a moderate temperature setting (T = 2.0) achieves the best balance between accuracy and perplexity in this experiment. At this value, the student model effectively absorbs knowledge from the teacher model while maintaining strong reasoning ability. In contrast, excessively low or high temperatures compromise distillation performance. Therefore, in practical applications, selecting and optimizing temperature parameters according to specific task requirements is essential for achieving optimal knowledge distillation.

Secondly, this paper also gives experiments on the influence of different levels of characteristic distillation on the performance of the student model, and the experimental results are shown in Figure 2.

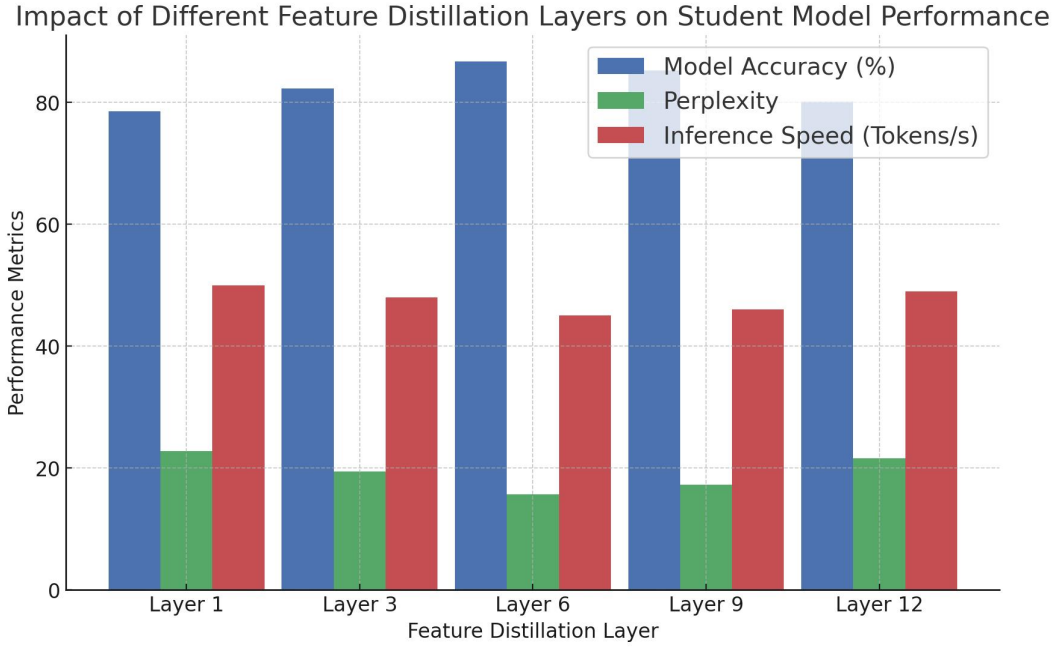

Figure 2. Impact of Different Feature Distillation Layers on Student Model Performance

According to the experimental results, different levels of feature distillation significantly impact student model performance. This effect is mainly reflected in three aspects: model accuracy, perplexity, and inference speed. First, model accuracy reaches its highest value at Layer 6 (approximately 87%). This suggests that mid-layer feature distillation provides the most effective information transfer, enabling the student model to achieve optimal classification performance. In contrast, accuracy is lower when distillation occurs at the shallowest (Layer 1) or deepest (Layer 12) layers. This indicates that early-layer distillation may provide insufficient information, while late-layer distillation may introduce excessive complexity, both of which negatively affect model performance.

In terms of perplexity, Layer 6 achieves the lowest value (around 15). This suggests that feature distillation at this layer results in better text generation, with outputs that exhibit higher fluency and semantic coherence. However, at Layer 1 and Layer 12, perplexity increases significantly, reaching approximately 22. This indicates greater uncertainty in the student model's text generation process, suggesting that it struggles to fully capture the teacher model's semantic representation. These findings

further confirm that mid-layer feature distillation efficiently compresses knowledge, allowing the student model to maintain a compact structure while preserving strong text generation capabilities.

Inference speed remains relatively stable across different layers, fluctuating between 45 and 50 tokens per second. However, Layer 6 has the slowest inference speed (approximately 45 tokens per second). This may be due to the larger volume of information involved in mid-layer distillation, requiring the student model to process more features during inference, leading to additional computational overhead. Despite this, Layer 6 achieves the highest accuracy and the lowest perplexity. Therefore, selecting a mid-layer for feature distillation (such as Layer 6) appears to be the most effective strategy for improving student model performance. At the same time, balancing inference efficiency and model effectiveness remains essential for practical applications.

Finally, this paper presents a model performance comparison experiment based on LoRA and full-parameter fine-tuning, and the experimental results are shown in Figure 3.

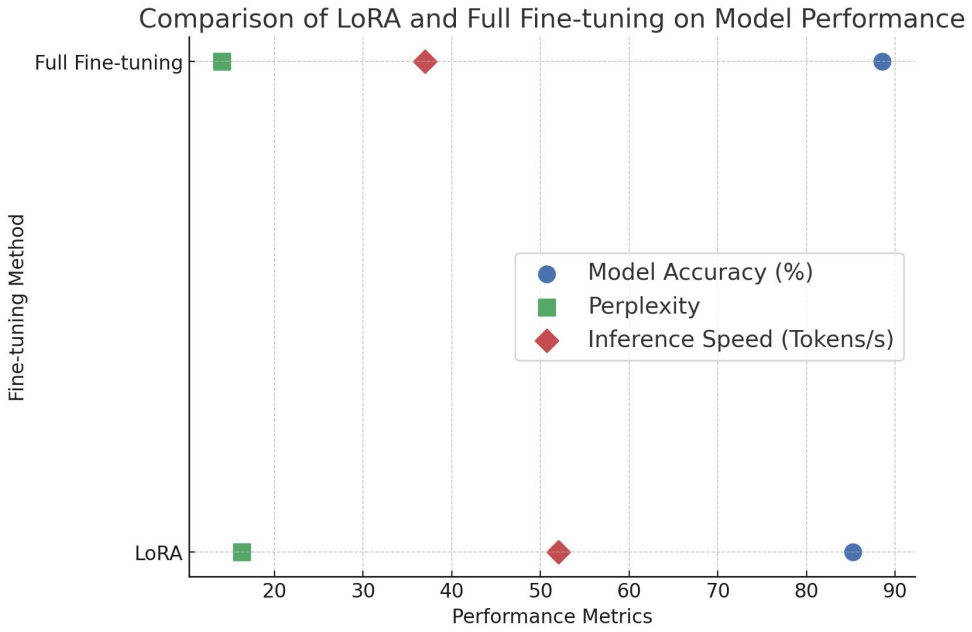

Figure 3. Comparison of LoRA and Full Fine-tuning on Model Performance

From the experimental results, there are obvious differences between Full Fine-tuning and LoRA in model performance and reasoning efficiency. Full-parameter fine-tuning has the highest Model Accuracy of 88.5%, compared to 85.2% for LoRA, indicating that the model can adjust the weights more fully when all parameters are trainable, thus better adapting to a specific task. However, this performance improvement comes at a higher computational cost, with the reasoning speed of full parameter fine-tuning being slower at only 37 tokens/s, while LoRA is significantly more computationally efficient at 52 tokens/s due to the adjustment of only a few low-rank matrices.

In terms of Perplexity, the perplexity of full-parameter fine-tuning was the lowest, which was 14.1, while that of LoRA was 16.3, indicating that the model after full-parameter fine-tuning was better in terms of language understanding and generation ability, and was able to generate more natural and smooth texts. Reduced confusion generally means that the model's output is more stable and the prediction is less

uncertain, so for tasks that require high accuracy, such as legal text analysis or medical text summaries, full-parameter fine-tuning may be a better choice. However, it should be noted that the confusion degree of LoRA is still in the acceptable range, indicating that it can still maintain good generation quality under the premise of low computing cost.

From an overall perspective, LoRA is suitable for computant-constrained scenarios, such as edge computing devices or fine-tuning tasks in low-resource environments, because it offers significant advantages in reasoning speed and storage efficiency, while full-parameter fine-tuning is suitable for tasks with high performance requirements, such as financial modeling or high-precision NLP applications. The choice of the two should be based on the specific application scenario to balance the computing cost and performance requirements, if the goal is to achieve better results under limited computing resources, LoRA is the better solution; If the computing resources are sufficient and the model accuracy is the ultimate pursuit, full-parameter fine-tuning is more appropriate.

4. Conclusion

In this paper, the optimization strategies combining knowledge distillation and fine-tuning techniques are discussed for the lightweight of large language models, and different methods are analyzed through experiments. The experimental results show that knowledge distillation can effectively compress the model size and improve the generalization ability of students' models through reasonable selection of temperature parameters and characteristic distillation layers. In terms of fine-tuning, LoRA, as a parametric efficient fine-tuning method, performs well in reasoning speed and computational cost, while full-parameter fine-tuning has more advantages in accuracy and language understanding ability. These results show that different optimization strategies have their own applicability in different application scenarios, and researchers need to weigh computing resources and model performance according to actual needs to choose a suitable lightweight scheme.

In addition, the experimental results also reveal the importance of the combination of distillation and fine-tuning, that is, by selecting the appropriate level characteristics in the distillation process and combining them with efficient fine-tuning strategies, the inference overhead can be greatly reduced while maintaining the model performance. Distillation of specific layers helps the student model more effectively inherit the knowledge of the teacher model, while parametric efficient fine-tuning further optimizes the model's adaptability to a specific task while maintaining a small number of trainable parameters. The combination of these technologies can not only improve the application capability of large language models in low-resource environments but also provide new ideas for industry to deploy efficient NLP systems.

Future research could further explore more advanced methods of knowledge distillation and fine-tuning, such as more refined knowledge transfer combined with contrast learning or further compression of models through structured pruning and mixed precision quantization. In addition, task-specific adaptive lightweight strategy is still a direction worthy of further research. How to find the optimal model compression strategy between different task requirements will be one of the key challenges in the lightweight research of large language models. The results of this study provide practical guidance for the application of large models in the computing power constrained environment and lay a foundation for future optimization methods.

References

[1] Sanh V, Debut L, Chaumond J, et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter[J]. arXiv preprint arXiv:1910.01108, 2019.

[2] Jiao X, Yin Y, Shang L, et al. TinyBERT: Distilling BERT for natural language understanding[J]. arXiv preprint arXiv:1909.10351, 2019.

[3] Sun Z, Yu H, Song X, et al. MobileBERT: a compact task-agnostic BERT for resource-limited devices[J]. arXiv preprint arXiv:2004.02984, 2020.

[4] Wang W, Wei F, Dong L, et al. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers[J]. arXiv preprint arXiv:2002.10957, 2020.

[5] Turc I, Chang M W, Lee K, et al. Well-read students learn better: On the importance of pre-training compact models[J]. arXiv preprint arXiv:1908.08962, 2019.

[6] Tang R, Lu Y, Liu L, et al. Distilling task-specific knowledge from BERT into simple neural networks[J]. arXiv preprint arXiv:1903.12136, 2019.

[7] Mukherjee S, Awadallah A H. Xtremedistil: Multi-stage distillation for massive multilingual models[J]. arXiv preprint arXiv:2004.05686, 2020.

[8] Liang Y, Wang Y, Zhang K, et al. MixKD: Towards efficient distillation of large-scale language models[J]. arXiv preprint arXiv:2011.02255, 2020.

[9] Zhao Y, Ni X, Ding Y, et al. ExtremeBERT: A toolkit for accelerating pretraining of customized BERT[J]. arXiv preprint arXiv:2004.07740, 2020.