Sparse Mixture of Experts Language Models Excel in Knowledge Distillation

Haiyang Xu[1], Haoxiang Liu[2], Wei Gong[1(✉)], Xianjun Deng[3], and Hai Wang[4]

[1] University of Science and Technology of China, Anhui, 230026, P.R.China
sprog_xhy@mail.ustc.edu.cn, weigong@ustc.edu.cn
[2] Alibaba Group
[3] School of Cyber Science and Engineering, Huazhong University of Science and Technology
dengxj615@hust.edu.cn
[4] School of Computer Science and Engineering, Southeast University
hai@seu.edu.cn

Abstract. Knowledge distillation is an effective method for reducing the computational overhead of large language models. However, recent optimization efforts in distilling large language models have primarily focused on loss functions and training methodologies, with limited attention given to structural improvements of student models. This is largely due to the challenges posed by cross-architecture distillation and the substantial computational resources required for modifying model structures. To address these issues, we introduce a novel method that integrates a sparse mixture of experts (MoE) architecture with low-rank adaptation (LoRA). This combination not only bolsters the capabilities of the student model but also facilitates knowledge distillation using MoE without the necessity of continued pretraining. Experimental results indicate that our approach enhances the model's capabilities compared to dense model distillation, achieving superior performance across a multitude of tasks. We will release our code.

Keywords: Mixture of experts · Knowledge distillation · Language models.

1 Introduction

Current auto-regressive large language models [18,24,32] have achieved significant success across various domains. However, their computational and memory requirements are substantial, which hinders the deployment of these models in practical scenarios. To mitigate the computational consumption of large language models, researchers have proposed numerous methods [8,17,23], among which knowledge distillation [16] stands out as a key technique. It compresses insights from a large teacher model into a smaller student model, thereby reducing computational costs.

While leveraging knowledge distillation to decrease the number of parameters in language models, researchers have also addressed the performance loss

issue by improving loss functions and training techniques. For instance, [15,34] have employed the f-divergence function or reverse Kullback-Leibler divergence (KLD) to refine the commonly used forward KLD loss in distillation. This allows the student model to better capture the distribution of the teacher model and enhances its generative capabilities [36]. Moreover, [1,19] have optimized the training process of knowledge distillation using reinforcement learning and training on outputs generated by the student model.

These distillation techniques have improved the effectiveness of model distillation by refining loss functions and training methods. However, they seldom focus on optimizing the structure of the student model, typically distilling from a dense teacher model [11] to a dense student model of the same architecture. In previous approaches, altering the student model's structure or using a stronger architecture is rarely considered [37]. On the one hand, changing the model structure necessitates continued pretraining [9,28], which requires computational resources equivalent to hundreds of times of fine-tuning [3]. Moreover, it is challenging to determine which structural modifications are more effective. On the other hand, distilling to a stronger pre-trained model with a different structure encounters difficulties [2] with the tokenizer.

Inspired by the exceptional performance of mixture-of-experts models [12,18], as shown in the Fig. 1, we propose distilling from a dense model to a stronger sparse mixture-of-experts model [31] to reduce the capability gap between the student and teacher models, thereby enhancing model performance. Building upon the work of [5,13,35], we have designed the low-rank mixture-of-experts model structure, which strengthens the student's capabilities without continued pretraining. We conducted experiments on three task-agnostic and three task-specific datasets. The results indicate that our method, termed MoE-KD, outperforms distillation to dense student models on five tasks.

Our contribution involves combining MoE with LoRA [17] to enhance the effectiveness of knowledge distillation across various tasks. Moreover, our method is naturally compatible with other distillation techniques [15,19].

2 Related Work

2.1 Sparse Mixture of Experts

In dense models, all distinct token inputs pass through all parameters of the model. In contrast, sparse models [11] route different token inputs through part of the model parameters. Consider a sparse model with N Multilayer Perceptron (MLP) networks, each serving as an expert E_i. A trainable weight matrix W_r of shape (d_h, N), where d_h is the dimension of the hidden states and N is the number of experts, acts as a router that selects expert outputs. For a token embedding or hidden state x of a certain layer, the router computes the matrix $h(x)$ as follow:

$$h(x) = W_r^{\mathrm{T}} x \tag{1}$$

The softmax function is then applied to $h(x)$ to obtain the weight $p_i(x)$ of each expert E_i:

$$p_i(x) = \frac{\exp(h(x)_i)}{\sum_{j=0}^{N} \exp(h(x)_j)} \tag{2}$$

The top K experts are selected based on $p_i(x)$ to form the set Γ and and the outputs of experts in Γ are summed, weighted by $p_i(x)$, to obtain the final output y:

$$y = \sum_{i \in \Gamma} p_i(x) E_i(x) \tag{3}$$

2.2 Knowledge Distillation

Knowledge distillation is a pivotal model compression technique that facilitates the transfer of knowledge from a large, intricate teacher model to a more compact student model. The overarching goal of knowledge distillation is to minimize the KLD denoted as $D_{\mathrm{KL}}(p, q_\theta)$ [15,29], between teacher's probability distribution p and student's probability distribution q_θ. For instance, when a text input x of length S is processed, the KLD loss function can be articulate as follows:

$$D_{\mathrm{KL}}(p, q_\theta) = \sum_{i=0}^{S} p(x_i) \log \frac{p(x_i)}{q_\theta(x_i)} \tag{4}$$

The KLD loss function quantifies the difference between these two distributions, guiding the student model to closely approximate the teacher's nuanced understanding of the input data.

2.3 Low-rank Adaptation

LoRA [17] is an efficient fine-tuning method for language models. In LoRA, the model's weight matrix is decomposed into two components: a low-rank matrix ΔW and a fixed base matrix W. The low-rank matrix captures changes during the fine-tuning process, while the base matrix retains the pre-trained model's original information. We define the base matrix W to have dimensions $(h_{\mathrm{input}}, h_{\mathrm{output}})$, where h_{input} and h_{output} represent the input and output feature space sizes respectively. The corresponding low-rank matrix ΔW is constructed by taking the product of two matrices, denoted as A and B.

$$y = Wx + \Delta W x = Wx + ABx \tag{5}$$

The matrix A has the shape (h_{input}, r), and matrix B has the shape (r, h_{output}), where $r \ll min(h_{\mathrm{input}}, h_{\mathrm{output}})$.

3 Methods

Our distillation method, named MoE-KD, comprises two key components: the mixture of low-rank experts and the mixture of losses.

3.1 Mixture of Low-rank Experts

The Mixture of Experts is a powerful model that combines the predictions of multiple expert models, each of which specializes in a different part of the input space. This is superior to a single MLP module as it allows for more efficient and effective use of model parameters.

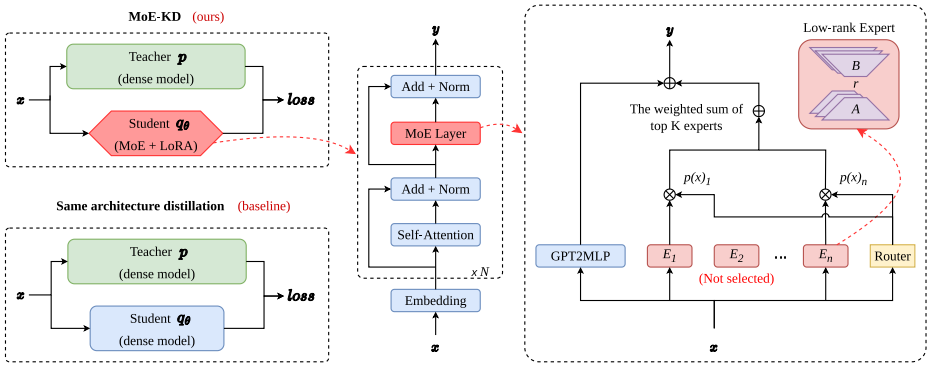

Fig. 1. Integration of MoE with LoRA in student model architecture.

We have successfully integrated MoE with the LoRA approach. Our method stands in contrast to that of [37], who directly convert the BERT [9] student model into an MoE structure. As depicted in Fig. 1, to circumvent retraining, we have not modified the backbone model's architecture. Instead, we introduce sparse LoRA experts that are strategically in parallel with the MLP of the GPT-2 [27] model. This innovative approach enables efficient integration of the MoE framework with the student model.

In each MoE layer, each low-rank expert is instantiated as a MLP module. We replaced each matrix W_{ij} in the low-rank expert E_i with two low-rank matrices A_{ij} and B_{ij} with a minimal number of parameters. We denote the LeakyReLU activation function as σ. Consequently, the output of the low-rank expert is represented as in equation (6).

$$E_i(x) = (A_{i2}B_{i2})^{\mathrm{T}}\sigma((A_{i1}B_{i1})^{\mathrm{T}}xA_{i3}B_{i3}) \tag{6}$$

The outputs of the low-rank experts sub-modules and the outputs of the backbone network $G(x)$ are summed to obtain the final output as shown in the equation (7).

$$y = G(x) + \sum_{i \in \Gamma} p_i(x) E_i(x) \tag{7}$$

Trained routing dynamically selects appropriate experts for each token, resulting in superior performance compared to a single-expert MLP layer.

Our method integrates seamlessly with the backbone network model and is compatible with various distillation techniques [15,19] that optimize loss and training methodologies. During training, we can choose to optimize only the low-rank experts module or perform full tuning.

3.2 Mixture of Losses

In our approach, we employ a mixture of losses L_{mix} to improve the distillation process and the model's performance on diverse tasks.

The first component of our loss function is the Skew Reverse Kullback-Leibler (SRKL) divergence, defined as equation (8). The SRKL divergence is used as our distillation loss function, with α being a hyperparameter that skews the divergence in favor of the teacher model's distribution.

$$D^{\alpha}_{\text{SRKL}}(p, q_\theta) = D_{\text{KL}}(q_\theta, (1-\alpha)p + \alpha q_\theta) \tag{8}$$

[19] has shown that SRKL is an effective loss function, achieving state-of-the-art results. We adopt SRKL to improve distillation and assess compatibility with existing techniques. Additionally, we incorporate a Supervised Fine-Tune (SFT) [25] loss into our total loss function to improve performance on diverse tasks. When a text input sequence x comprising S tokens is processed, the SFT loss is computed as equation (9), where $P(x_i|x_{<i})$ denotes the probability assigned by the model to the i-th token in the x.

$$L_{\text{SFT}} = -\sum_{i=1}^{S} \log P(x_i|x_{<i}) \tag{9}$$

We use β to control the weight of SFT loss in the total loss L_{mix} as in equation (10). In our experiments, we set β to 0.1, a value determined through preliminary experiments.

$$L_{\text{mix}} = D^{\alpha}_{\text{SRKL}}(p, q_\theta) + \beta L_{\text{SFT}} \tag{10}$$

The SRKL loss facilitates students' learning of the prior distribution over teacher models, while the SFT loss enables the model to better learn the inherent distribution of the dataset. This mixture of losses provides a balance between the distillation process and the model's performance on the downstream task.

4 Experiments

This section presents a thorough evaluation of our proposed methods, demonstrating their efficacy on both task-specific and task-agnostic data. We base our evaluation on the work of [15,19] and apply our methods to open question answering, text summarization, and machine translation (English to Chinese) tasks.

4.1 Task-Specific Distillation

We conducted task-specific distillation experiments on three datasets: Natural Questions (NQ) [20] for open question answering, SAMSum [14] for text summarization, and IWSLT 2017 En-Zh [4] for machine translation. Evaluation metrics comprised Exact Match for NQ to ensure absolute textual agreement, ROUGE-L [22] for SAMSum to evaluate summary content alignment, effectively balancing precision and recall, and BLEU [26] for IWSLT to assess both translation quality and fluency.

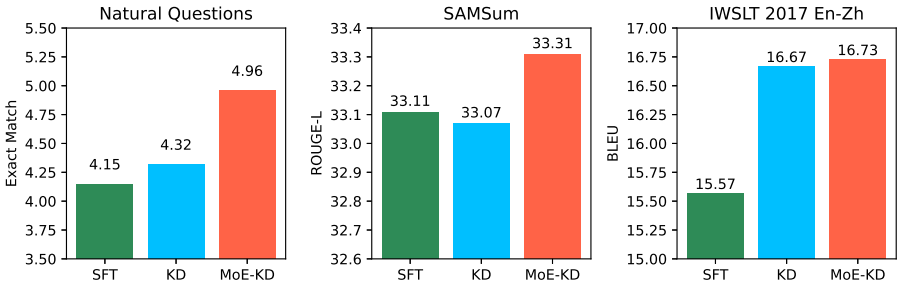

Fig. 2. Comparison of model performance on three task-specific datasets

We denote distillation to a dense model as KD, which serves as our baseline. As shown in Fig. 2, our method, MoE-KD, outperformed KD baseline on three tasks. In distillation, all variables in MoE-KD and KD are the same, except for the model structure. However, it can be seen that MoE-KD outperforms dense model distillation by 0.64, 0.24, and 0.06 on three tasks, respectively. These results suggest that the MoE architecture is beneficial for distillation on task-specific datasets.

In the context of our experimental setup, we first supervised fine-tuned GPT-2-large [27] and GPT-2-small on three datasets for 10 epochs, using the large SFT model as the teacher model and the small SFT model as the student model. Both the knowledge distillation and MoE models were initialized from the small SFT model. The sparse MoE model was obtained by adding 2 to 8 experts with a rank of 16 to the last 6 layers of GPT-2-small. Then we distilled knowledge to

dense models and MoE models for 5 epochs and evaluated the best 3 checkpoints. Furthermore, we applied SFT training to the MoE model on the NQ dataset, creating a control group called MoE-SFT.

4.2 Task-Agnostic Distillation

We train models on Dolly [7] and evaluate on Dolly, Self-Inst [33], Vicuna [6] datasets based on the work of [19]. The training process for SFT and knowledge distillation is the same as the task-specific distillation process described in the previous chapter, but the MoE model structure differs. The sparse MoE model was obtained by adding 8 experts with a rank of 16 to all layers of GPT-2-small.

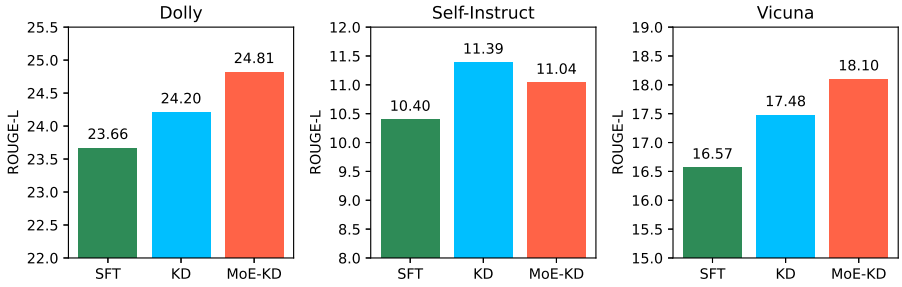

Fig. 3. Comparison of model performance on three task-agnostic datasets

As shown in Fig. 3, we adopted ROUGE-L as our evaluation metric. Our method, MoE-KD, outperformed the KD baseline on the Dolly and Vicuna datasets, but performed worse on the Self-Instruct dataset.

Additionally, we compared our method with other distillation techniques, including those that were improved by loss and training methodologies. The experimental results, shown in Table 1, indicate that our method exhibits comparable performance to the latest methods. Our approach is compatible with the other methods listed in the table; however, due to limited experimental resources, we did not integrate it with them. For instance, the MiniLLM [15] work utilized the resource-intensive PPO [30] algorithm, which exceeded our resource capabilities. Nevertheless, it is evident that the enhancement of student model performance through MoE leads to a significant improvement in distillation effectiveness.

4.3 Configuration of Experts

We conducted a systematic investigation into key parameters of low-rank experts' configuration, as they significantly impact the efficacy and computational demands of knowledge distillation. As shown in Table 2, these parameters include the LoRA rank r, the MoE layer placement and the number of experts.

Table 1. Comparison of ROUGE-L Scores for Different Distillation Methods on the Dolly, Self-Instruct, and Vicuna Datasets

Method	Year	Dolly	Self-Instruct	Vicuna
Vanilla KD [16]	2015	23.52	11.23	15.92
ImitKD [21]	2020	21.63	10.85	14.70
MiniLLM [15]	2024	<u>23.84</u>	<u>12.44</u>	**18.29**
GKD [1]	2024	23.75	**12.73**	16.64
MoE-KD (ours)	2024	**24.81**	11.04	<u>18.10</u>

We only incorporate MoE layers in the final few layers of the model. All models were trained and evaluated on the Dolly dataset.

Table 2. Performance comparison of different experts' configuration settings

Rank r	MoE layer count	Number of experts	ROUGE-L on Dolly	Training speed (seconds/step)
4	6	8	23.83	0.890
16	6	8	24.03	0.877
64	6	8	23.96	0.877
256	6	8	**24.04**	0.879
16	12	8	23.91	1.118
16	6	8	**24.03**	0.877
16	3	8	23.81	0.756
16	1	8	23.87	0.695
16	6	2	23.96	0.803
16	6	8	23.85	0.964
16	6	32	**24.13**	1.267
16	6	128	23.91	2.203

We designed experiments to vary the count of experts per layer, incrementing from 2 to 128 experts in a stepwise manner. For expert placement, we positioned a higher number of expert layers at the upper layers of the model, following the approach of [13]. For the 12-layer GPT-2-small model, we explored configurations with 12, 6, 3, and 1 expert layer(s) to assess the impact on distillation performance.

To determine the optimal setting for the LoRA rank r, we tested values ranging from 4 to 256. We also monitored the training duration for each configuration to evaluate the trade-off between computational efficiency and distillation effectiveness.

5 Analysis

5.1 Capabilities of MoE models

As shown in Table 3, The results reveal that sparse MoE models have stronger capabilities than their corresponding dense models. When all other conditions are held constant, the MoE model trained with SFT outperforms the dense model, and the MoE model trained with knowledge distillation also outperforms the dense model. For example, on the NQ dataset, MoE-SFT's Exact Match metric improved by 0.56 compared to SFT, and MoE-KD's Exact Match improved by 0.64 compared to KD. This demonstrates that by adding LoRA experts, we have enhanced the student model's capabilities.

Table 3. The performance gap between teacher and student models on six tasks.

Datasets	Metrics	Teacher	SFT	MoE-SFT	KD	MoE-KD
NQ	Exact Match	14.94	4.15	<u>4.71</u>	4.32	**4.96**
NQ	ROUGE-L	25.57	11.58	<u>12.28</u>	11.91	**12.55**
SAMSum	ROUGE-L	34.95	33.11	-	33.07	**33.31**
IWSLT 2017	BLEU	18.06	15.57	-	16.67	**16.73**
Dolly	ROUGE-L	26.82	23.66	24.13	<u>24.20</u>	**24.81**
Self-Instruct	ROUGE-L	13.15	10.40	9.86	**11.39**	<u>11.04</u>
Vicuna	ROUGE-L	20.04	16.57	<u>17.79</u>	17.48	**18.10**

However, the MoE model performs worse than the dense model on the Self-Instruct dataset. This may be due to a distributional gap between the dolly dataset used for training and the Self-Instruct dataset used for evaluation. Additionally, the MoE model may have overfit to the dolly dataset, which could have weakened its generalization ability on the Self-Instruct task.

5.2 Impact of Experts' Configuration

The impact of experts' configuration on model performance is complex. As shown in Table 2, when the rank of experts exceeds 16, there is no further improvement in model performance. The optimal number of layers to place experts is in the last 6 layers, which performs even better than adding experts to every layer. The best performance is achieved when adding 32 experts to each layer.

Experts' configuration significantly affects model training time. The rank of low-rank experts has little impact on training time. However, increasing the number of experts and the number of layers they are placed on will significantly increase training time, especially the number of experts.

6 Conclusion

We introduce the MoE and LoRA structure into knowledge distillation, and improve the effect of knowledge distillation by reducing the gap between the

student model and the teacher model, which provides a novel perspective for researchers to improve knowledge distillation of language models. Our method MoE-KD is naturally compatible with methods [15,19,34] that improve knowledge distillation from loss and training, and our method outperforms distillation to dense models on multiple tasks.

The deficiency of our work lies in the lack of load balancing loss [10,12] for low-rank experts and the lack of optimization for the training time growth brought by MoE, which will be the direction of our future work.

Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments and suggestions.

References

1. Agarwal, R., Vieillard, N., Zhou, Y., Stanczyk, P., Garea, S.R., Geist, M., Bachem, O.: On-policy distillation of language models: Learning from self-generated mistakes. In: The Twelfth International Conference on Learning Representations (2024)
2. Boizard, N., El-Haddad, K., Hudelot, C., Colombo, P.: Towards cross-tokenizer distillation: the universal logit distillation loss for llms. arXiv preprint arXiv:2402.12030 (2024)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
4. Cettolo, M., Federico, M., Bentivogli, L., Niehues, J., Stüker, S., Sudoh, K., Yoshino, K., Federmann, C.: Overview of the iwslt 2017 evaluation campaign. In: Proceedings of the 14th International Workshop on Spoken Language Translation. pp. 2–14 (2017)
5. Chen, S., Jie, Z., Ma, L.: Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. arXiv preprint arXiv:2401.16160 (2024)
6. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023) **2**(3), 6 (2023)
7. Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., Xin, R.: Free dolly: Introducing the world's first truly open instruction-tuned llm. Company Blog of Databricks (2023)
8. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. Advances in Neural Information Processing Systems **36** (2024)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

10. Dikkala, N., Ghosh, N., Meka, R., Panigrahy, R., Vyas, N., Wang, X.: On the benefits of learning to route in mixture-of-experts models. In: The 2023 Conference on Empirical Methods in Natural Language Processing (2023)
11. Fedus, W., Dean, J., Zoph, B.: A review of sparse expert models in deep learning. arXiv preprint arXiv:2209.01667 (2022)
12. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research **23**(120), 1–39 (2022)
13. Gao, C., Chen, K., Rao, J., Sun, B., Liu, R., Peng, D., Zhang, Y., Guo, X., Yang, J., Subrahmanian, V.: Higher layers need more lora experts. arXiv preprint arXiv:2402.08562 (2024)
14. Gliwa, B., Mochol, I., Biesek, M., Wawer, A.: Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. arXiv preprint arXiv:1911.12237 (2019)
15. Gu, Y., Dong, L., Wei, F., Huang, M.: Minillm: Knowledge distillation of large language models. In: The Twelfth International Conference on Learning Representations (2023)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
17. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
18. Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., Casas, D.d.l., Hanna, E.B., Bressand, F., et al.: Mixtral of experts. arXiv preprint arXiv:2401.04088 (2024)
19. Ko, J., Kim, S., Chen, T., Yun, S.Y.: Distillm: Towards streamlined distillation for large language models. arXiv preprint arXiv:2402.03898 (2024)
20. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al.: Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics **7**, 453–466 (2019)
21. Lin, A., Wohlwend, J., Chen, H., Lei, T.: Autoregressive knowledge distillation through imitation learning. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Nov 2020)
22. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-1013
23. Ma, S., Wang, H., Ma, L., Wang, L., Wang, W., Huang, S., Dong, L., Wang, R., Xue, J., Wei, F.: The era of 1-bit llms: All large language models are in 1.58 bits. arXiv preprint arXiv:2402.17764 (2024)
24. OpenAI, R.: Gpt-4 technical report. arxiv 2303.08774. View in Article **2**(5) (2023)
25. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
26. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
27. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

28. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)
29. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
30. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
31. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
32. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
33. Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N.A., Khashabi, D., Hajishirzi, H.: Self-instruct: Aligning language models with self-generated instructions. arXiv preprint arXiv:2212.10560 (2022)
34. Wen, Y., Li, Z., Du, W., Mou, L.: f-divergence minimization for sequence-level knowledge distillation. arXiv preprint arXiv:2307.15190 (2023)
35. Wu, X., Huang, S., Wei, F.: Mole: Mixture of lora experts. In: The Twelfth International Conference on Learning Representations (2023)
36. Xu, X., Li, M., Tao, C., Shen, T., Cheng, R., Li, J., Xu, C., Tao, D., Zhou, T.: A survey on knowledge distillation of large language models. arXiv preprint arXiv:2402.13116 (2024)
37. Zhang, C., Yang, Y., Liu, J., Wang, J., Xian, Y., Wang, B., Song, D.: Lifting the curse of capacity gap in distilling language models. arXiv preprint arXiv:2305.12129 (2023)