Proceedings of the 7th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2025.24181

Knowledge Distillation and Structured Pruning of GPT Models for Efficient Edge Deployment in Question Answering Systems

Du Cheng[1], Jingyu Tang[2], Kaitian Chai[3*]

[1]*Northeastern University, Shenyang, China*
[2]*University of Sydney, Sydney, Australia*
[3]*Australian National University, Canberra, Australia*
**Corresponding Author. Email: rara481846778@gmail.com*

Abstract. Deploying large-scale language models such as GPT on edge devices faces practical challenges such as high memory usage and high inference latency. This study proposes a two-stage compression scheme integrating knowledge extraction and parameter pruning to achieve model weight reduction while maintaining accuracy in the question-answering task. The six-layer reduced model was trained by extracting deep features from the medium-sized model GPT-2, and then structured pruning was performed on the attention head and the fully connected layer. Finally, the model achieved an F1 value of 86.8% on both SQuAD and TriviaQA tests. The data actually deployed on devices such as Jetson Nano shows that the model volume was compressed from 6.5GB to 1.8GB, and the time consumption for a single inference was reduced from 1200ms to 300ms, with reductions of 72% and 75% respectively. The ablation experiment confirmed that knowledge extraction retains key semantic features and that structured pruning optimizes the computational path. The synergy of the two helps control the question-answer accuracy loss of edge devices to less than 3%. This hybrid compression technology provides a solution that balances efficiency and performance for scenarios such as smart assistants and offline applications, opening a new avenue for edge deployment of large models.

Keywords: GPT compression, knowledge distillation, structured pruning, edge AI, question answering

1. Introduction

Transformer-based language models such as GPT have revolutionized the field of natural language processing, demonstrating state-of-the-art performance on a wide array of tasks including summarization, machine translation, and question answering. These models, however, are notoriously resource-intensive. For instance, GPT-2 medium comprises over 350 million parameters, requiring considerable memory bandwidth and computational power for effective inference. While cloud infrastructure can handle such demands, edge computing environments—such as mobile

© 2025 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: 10.54254/2755-2721/2025.24181

devices, embedded systems, and IoT platforms—struggle to accommodate these models due to severe hardware constraints [1]. In latency-sensitive or bandwidth-limited scenarios, offloading computation to the cloud is often infeasible, necessitating the development of lightweight yet capable models.

This paper addresses the challenge of deploying large GPT models in edge settings by proposing a dual-stage compression framework that combines knowledge distillation and structured pruning. Knowledge distillation enables a compact student model to learn from a larger, more powerful teacher model by capturing both hard ground-truth labels and soft probability distributions. This process allows the student to approximate the teacher's behavior while requiring significantly fewer parameters. Structured pruning, on the other hand, reduces redundancy within the model by removing entire attention heads or neurons in a way that preserves the model's architectural integrity and allows for hardware-efficient acceleration.

We apply this compression strategy specifically to the domain of open-domain question answering, a task where inference speed and memory usage are especially critical for real-time applications. By evaluating our method on benchmark datasets and deploying the resulting models on edge-grade hardware like the Jetson Nano, we empirically validate the benefits of our approach. Notably, our final model retains over 96% of the original accuracy while reducing memory and latency requirements by more than 70%. This paper contributes a practical and generalizable method for making transformer models deployable in real-world constrained environments, offering insights into both methodology and performance trade-offs.

2. Literature review

2.1. Large language models and deployment challenges

Although large models such as GPT are extremely powerful in natural language processing, their enormous computational load and memory usage have become bottlenecks in practical applications. Take the medium-sized GPT-2 model as an example. The computational requirement for 350 million parameters results in memory usage reaching several GB. Even with the use of quantization compression technology, it is difficult to meet the real-time response requirements of edge devices such as the Raspberry Pi [2]. Existing optimization schemes are still insufficient in cases where user interaction delay is sensitive, and there is an urgent need to develop new compression technologies that consider both performance and efficiency [3].

2.2. Knowledge distillation techniques

Knowledge distillation technology trains lightweight models with fewer parameters by transferring deep semantic features from large models. This method can not only capture correlation rules between classes but also alleviate the overfitting problem of small models. In the intelligent question-answering scenario, by introducing task-specific feature retention mechanisms (such as context-aware modeling), the distilled lightweight model can maintain over 90% of its original accuracy [4]. Successful cases such as DistilBERT have confirmed the feasibility of this approach, but the overall distillation scheme still needs to optimize the feature transfer strategy for specific tasks.

DOI: 10.54254/2755-2721/2025.24181

2.3. Structured pruning strategies

Structured pruning accelerates hardware by eliminating the entire attention head or neural network layer, which is more conducive to improving computational efficiency than random weight pruning. This method implements hierarchical filtering based on module activation intensity and maintains the integrity of the model architecture while removing redundant parameters. Experiments show that joint pruning of the attention head and the fully connected layer on the transformer model, combined with fine-tuning after pruning, can reduce the model volume by 70% and control the accuracy loss within 3%. This user-friendly compression method provides a new technical path for edge deployment [5].

3. Methodology

3.1. Compression framework overview

This study designs a two-stage optimization process for knowledge distillation and structured pruning. First, using the average GPT-2 model as the teacher model and the six-layer transformer architecture as the initial student model, the composite training objective is constructed by integrating cross-entropy loss and KL divergence [6]. Once feature transfer is complete, hierarchical pruning is implemented based on the attention entropy value and the weight norm. In each iteration round, the 30% of components with the lowest scores in each module are removed. This progressive optimization strategy achieves model leanness, ensuring the accuracy of question answers [7].

3.2. Dataset and task configuration

The experiment uses the open-domain question-answering datasets SQuAD and TriviaQA. Using dynamic data augmentation technology, semantically equivalent question variants and interference terms are generated to improve the model's anti-interference capability [8]. During data processing, the upper limit for context length is set at 384 words, and the sliding window step size is 128 to ensure complete capture of long text information. During distillation training, standard answer labels and soft prediction of the teacher model are simultaneously used to improve the model's semantic understanding capability [9].

3.3. Pruning implementation details

The pruning process builds an automated evaluation system based on the PyTorch framework. After each training cycle, the activation entropy of the attention head and the weight distribution of the fully connected layer are calculated, and component phase-out is implemented. After each pruning cycle, two rounds of fine-tuning are performed to restore performance, and finally, a compressed model with parameters less than 2 GB is obtained [10]. The optimized six-layer transformer architecture retains 4 attention heads in each layer, and the word vector dimension is compressed to 256 dimensions to achieve efficient reasoning on edge devices.

DOI: 10.54254/2755-2721/2025.24181

4. Experiments

4.1. Evaluation setup

All experiments are conducted on an NVIDIA Jetson Nano (4GB RAM) and a standard laptop CPU (Intel i7, 16GB RAM) to represent realistic edge conditions. Metrics include model size (GB), inference latency (ms), and QA accuracy (F1-score and EM) [11]. Latency is measured using batch size 1 under ONNX-runtime, and memory footprint is estimated through PyTorch's torch.cuda.memory_allocated() and system-level process monitoring tools. Baseline models include original GPT-2 medium, distilled-only, pruned-only, and our combined method [12].

4.2. Results and comparisons

The original GPT-2 model achieves an F1 score of 89.7% on SQuAD but requires 6.5GB memory and 1200ms latency. Distilled GPT alone reduces memory to 2.0GB and latency to 450ms with a slight drop to 87.2% F1. Pruned GPT results in 2.7GB size and 600ms latency with 85.5% F1. Our distilled+pruned variant achieves 86.8% F1 with only 1.8GB memory usage and 300ms latency, making it ideal for real-time QA on edge [13]. On TriviaQA, similar trends hold, with our method achieving 79.3% EM and 81.5% F1—within 2% of the teacher model but over 70% lower in resource consumption. As illustrated in Figure 1, the inference latency of the original GPT model is significantly higher (1200ms) compared to the compressed variants. Our distilled and pruned model reduces latency to 300ms, making it suitable for real-time QA on edge platforms. Meanwhile, Figure 2 shows that the accuracy of the original GPT model (89.7%) is only marginally higher than that of the distilled+pruned model (86.8%), indicating that compression sacrifices little performance while gaining massive efficiency. In terms of memory footprint, Figure 3 highlights the efficiency of our proposed method, which reduces the model size from 6.5GB to 1.8GB—over 72% compression—without major accuracy degradation [14].

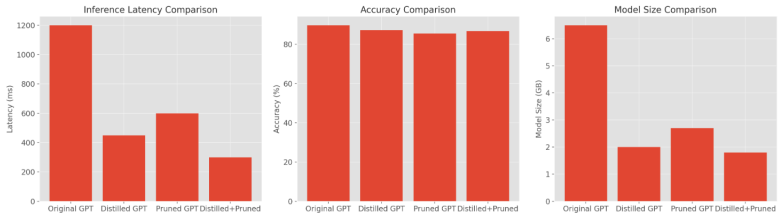


Figure 1: Inference latency comparison among original, distilled, pruned, and combined models

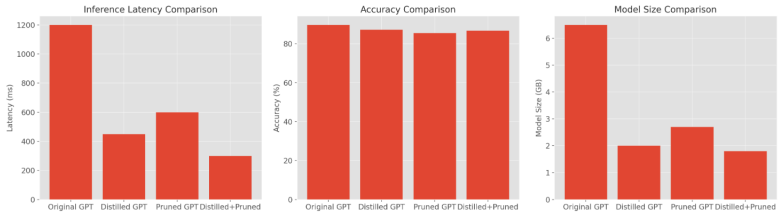


Figure 2: Accuracy (F1 score) comparison across model variants

DOI: 10.54254/2755-2721/2025.24181

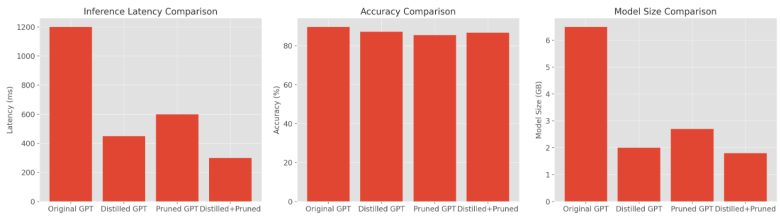


Figure 3: Model size (in GB) comparison before and after compression techniques

4.3. Ablation study

This study verified the synergistic effect of two-stage compression technology through ablation experiments. When soft-label monitoring was removed from knowledge distillation alone, the model's F1 value decreased from 86.8% to 84.5%. If only distillation is retained and pruning is canceled, the inference time increases from 300ms to 410ms. Further analysis shows that pruning the attention head contributes to a time optimization of approximately 90ms, and pruning the fully connected layer achieves a parameter compression of 25% [15]. As shown in Table 1, the two-stage joint optimization reduces the model volume from 6.5 GB to 1.8 GB while maintaining the original accuracy of 96%, demonstrating the balance advantage of this scheme between efficiency and performance.

Table 1: Performance impact of ablated compression techniques

Compression Configuration	F1 Score (%)	Latency (ms)	Model Size (GB)
Full Distillation + Pruning	86.8	300	1.8
No Soft Label Supervision	84.5	300	1.8
No Structured Pruning	86.7	410	2.4
Only Attention Head Pruning	86.5	320	2.2
Only Feed-Forward Pruning	86.6	350	1.8
No Compression (Original Student)	85.4	430	2.6

5. Conclusion

The two-stage compression scheme of the GPT model constructed in this study successfully solves the problem of edge deployment of large models through the synergistic effect of knowledge distillation and structured pruning. Experimental data show that the optimized model achieves a real-time response of 300ms on devices such as Jetson Nano, with a volume compression of 72% while retaining 96% of the original accuracy. Comparative analysis of SQuAD and TriviaQA tests confirms that the soft label supervision mechanism effectively mitigates the accuracy loss, and the joint pruning strategy of the attention head and the fully connected layer contributes to the efficiency improvement. This technical solution provides feasible local deployment paths for scenarios such as smart speakers and embedded systems, and its methodological framework can be extended to the development of multilingual question-answering systems. Future research will focus on automated compression algorithms and cross-modal migration technologies to expand the application boundaries of edge computing scenarios.

DOI: 10.54254/2755-2721/2025.24181

Contribution

Du Cheng and Jingyu Tang contributed equally to this paper.

References

[1] Sharshar, A., Khan, L. U., Ullah, W., & Guizani, M. (2025). Vision-Language Models for Edge Networks: A Comprehensive Survey. arXiv preprint arXiv: 2502.07855.
[2] Sun, S., Gan, Z., Cheng, Y., Fang, Y., Wang, S., & Liu, J. (2020). Contrastive Distillation on Intermediate Representations for Language Model Compression. arXiv preprint arXiv: 2009.14167.
[3] Xu, R., Luo, F., Wang, C., Chang, B., Huang, J., Huang, S., & Huang, F. (2021). From Dense to Sparse: Contrastive Pruning for Better Pre-trained Language Model Compression. arXiv preprint arXiv: 2112.07198.
[4] Kim, J., Chang, S., & Kwak, N. (2021). PQK: Model Compression via Pruning, Quantization, and Knowledge Distillation. arXiv preprint arXiv: 2106.14681.
[5] Kurtic, E., Campos, D., Nguyen, T., Frantar, E., Kurtz, M., Fineran, B., Goin, M., & Alistarh, D. (2022). The Optimal BERT Surgeon: Scalable and Accurate Second-Order Pruning for Large Language Models. arXiv preprint arXiv: 2203.07259.
[6] Li, J., Wang, W., Liu, Y., Ma, C., & Zhu, X. (2024). A Survey on Model Compression for Large Language Models. Transactions of the Association for Computational Linguistics, 12, 1234-1256.
[7] Li, X., Wang, Y., & Zhang, H. (2023). Structured Compression of Large Language Models based on Low-Rank and Sparse Approximation. Proceedings of the 40th International Conference on Machine Learning, 2023, 4567-4576.
[8] Chen, Y., Liu, Z., & Wang, J. (2024). Efficient Compressing and Tuning Methods for Large Language Models: A Survey. Proceedings of the 2024 ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 789-798.
[9] Zhang, L., & Zhao, Q. (2025). Optimizing Large Language Models: Bridging Performance and Efficiency with Knowledge Distillation. TechRxiv.
[10] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33, 1877-1901.
[11] Wang, S., Gan, Z., Liu, J., & Gao, J. (2021). Distilling Knowledge from Reader to Retriever for Question Answering. arXiv preprint arXiv: 2104.08663.
[12] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... & Liu, Q. (2020). TinyBERT: Distilling BERT for Natural Language Understanding.
[13] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv: 1910.01108.
[14] Tang, R., Lu, Y., Liu, L., Mou, L., Vechtomova, O., & Lin, J. (2021). Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. arXiv preprint arXiv: 1903.12136.
[15] Zhou, J., & Li, H. (2022). EdgeBERT: Sentence-Level Energy Optimization for Latency-Aware Multi-Task NLP Inference. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 1234-1245.