Journal of Industrial Engineering and Applied Science
ISSN 3005-6071 (Print) | ISSN 3005-608X (Online) | Vol. 2, No. 2, 2024

Research on Optimizing Lightweight Small Models Based on Generating Training Data with ChatGPT

DING, Rui [1*] ZHU, Elly yijun [2] ZHAO, Chao [3] YANG, Haoyu [3] LI, Jing [2] WU, Yue [2]

[1] San Francisco Bay University, USA
[2] Independent Researcher, USA
[3] Georgia Institute of Technology, USA

**** DING, Rui is the corresponding author, E-mail: rding267@student.sfbu.edu***

Abstract: This study aims to explore a method for optimizing lightweight small models in the field of deep learning by leveraging large models to generate training data. By introducing large-scale pre-trained models and enriching the training set through data generation, we enhance the performance of small models. The experimental results indicate that this strategy not only effectively improves the accuracy of lightweight models but also reduces computational expenses in resource-constrained environments.

Keywords: Deep Learning Optimization, Lightweight Model Enhancement, Computation Efficiency

DOI: https://doi.org/10.5281/zenodo.10841043

1 Introduction

ChatGPT is constructed upon OpenAI's robust large language models with targeted fine-tuning. In addition to engaging in conversational question answering, ChatGPT exhibits versatility in executing numerous natural language tasks, encompassing, but not restricted to, summarization, parsing unstructured text, text classification, translation, coding, and transformation. Capitalizing on the potent models and capabilities embedded in ChatGPT can substantially enhance the effectiveness of our AI-based features, products, and solutions. A direct and effective approach involves entrusting ChatGPT to seamlessly handle specific natural language tasks throughout the online inference process. But there are still many challenges and pain points to achieve that.

1. High expense: Implementing direct online usage often incurs significant expenses, primarily driven by the considerable compute cost of ChatGPT.

2. High latency: The response time for the publicly released ChatGPT typically hovers around several seconds. The latency rate is inadequate for tasks that demand millisecond-level responsiveness, making it unsuitable for time-sensitive operations.

3. Limited compute resources: The increasing deployment of ChatGPT-embedded tasks online poses a challenge as current computing resources, such as GPUs, are at risk of swift exhaustion.

4. Heavy upgrade effort: Integrating ChatGPT into existing online natural language tasks is a complex undertaking that goes beyond merely replacing a backend model. It necessitates considerations like changing deployment environments, rewriting core components, redesigning offline/online experiments, and incurring additional substantial effort, which could significantly impede progress.

ChatGPT, a formidable language generation model designed for versatile applications, faces challenges in direct online deployment due to high computational costs. This project explores a viable solution to transfer ChatGPT's capabilities to lightweight classical models, tailored for specific purposes, thereby achieving ChatGPT's prowess with cost-effectiveness.

Our proposed approach harnesses ChatGPT's zero-shot learning to train lightweight models possessing comparable capabilities, facilitating economical online inference. To elucidate, we draw parallels with the concept of "Knowledge Distillation (KD)".

The primary goal is to transfer knowledge from a large model (teacher), such as ChatGPT, to a small model (student) without compromising validity.

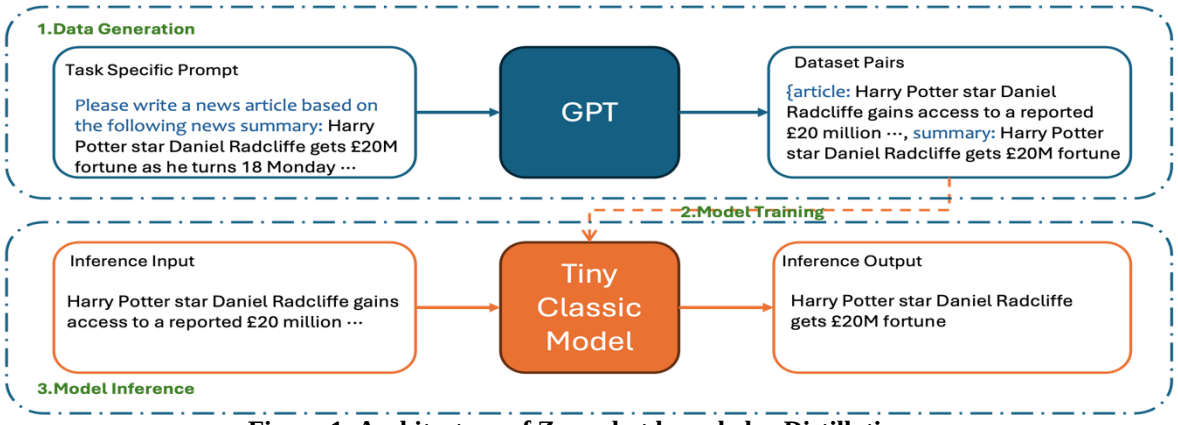

Figure 1. Architecture of Zero-shot knowledge Distillation.

In this context, ChatGPT serves as the teacher model, endowed with extensive knowledge capacity. The student model is typically a lightweight classical model, including GBDT (Friedman, 2001), LSTM (Hochreiter and Schmidhuber, 1997; Dai et al., 2023), BERT (Devlin et al., 2018), T5(Raffel et al., 2019), BART(Lewis et al., 2019), offering a cost-effective alternative for efficient online deployment.

This approach also holds promise in addressing the challenges associated with manual data labeling for supervised model training. In numerous natural language tasks within AI features, products, and solutions, specifically labeled datasets are often required for effective backend supervised machine learning models to support online inference. The quality and coverage of these labeled datasets directly influence the efficacy of the final trained models. Our approach mitigates the burden of manual data labeling by leveraging ChatGPT's zero-shot capability to automatically generate pseudo labels on real or predefined data without human involvement.

2 Related Work

Rather than relying on a substantial volume of annotated training data to refine Pretrained Language Models (PLMs) for downstream tasks, few-shot learning explores the optimal utilization of limited task-specific training data, a scenario more reflective of real-world applications. In the most stringent few-shot learning conditions, there is no assumption of access to unlabeled data or extensive validation sets for hyperparameter tuning (Perez et al., 2021). Prompt-based methods (Brown et al.,2020; Schick and Schütze, 2021; Tam et al., 2021; au2 et al., 2021) are prominently employed in such cases to infuse task descriptions into PLMs, leveraging their language modeling capability for enhanced training data efficiency, especially in low-data settings.

Expanding the scope, semi-supervised learning also makes use of unlabeled task-specific data (Liu et al., 2023; Li et al., 2024), employing common methods such as data augmentation, regularization (Miyato et al., 2018), and bootstrapping (Schick and Schütze, 2021). In contrast, zero-shot learning poses an even more formidable challenge by completely barring access to any task-specific data. When prompt-based methods are directly applied to extract predictions from PLMs without prior training, their zero-shot performance may significantly deteriorate (Brown et al., 2020). Formulating difficult Natural Language Understanding (NLU) tasks as prompts resembling the pretraining data format becomes a considerable hurdle, making it challenging for PLMs to accurately interpret and leverage the prompts in the absence of any training samples.

The prevailing trend in zero-shot learning centers around transfer learning: tasks with abundant annotations are converted into instruction templates (Mishra et al., 2022; Sanh et al., 2022; Xu et al., 2022), entailment pairs (Yin et al., 2019), or question-answer formats (Puri and Catanzaro,2019). By fine-tuning PLMs on these converted tasks, the PLMs acquire cross-task transferability (Ye et al., 2021), enabling them to perform unseen tasks when formulated in a similar format.

While previous research has predominantly focused on optimizing lightweight models through transfer learning, knowledge distillation, and other methods, these approaches are still constrained by the quantity and quality of training data. This study breaks through these limitations by introducing large pre-trained models and expanding the training set through data generation, thereby enhancing results and providing a more robust foundation for future advancements.

3 Method

The overall construction is shown as 1. It mainly consisted of 3 parts:

Data Generation: Use ChatGPT to generate dataset pair.

Tiny Classic Model Training: Use ChatGPT generated dataset to train the tiny classic model.

Model Inference: Use the trained tiny model to make the inference.

For a clearer presentation, this section will be based on

the experimental summarization task.

3.1 Summarization Task

Text summarization is a natural language processing task that involves generating concise and coherent summaries from longer documents or articles. The goal is to distill the essential information while maintaining the key points and overall meaning of the original content. This task is crucial for various applications, including information retrieval, document indexing, and content summarization for news articles, research papers, and more.

The CNN/Daily Mail dataset (See et al., 2017; Hermann et al., 2015) is a widely used corpus for training and evaluating text summarization models. It consists of news articles paired with multi-sentence summaries, providing a diverse and challenging set of examples for summarization tasks. The dataset is commonly employed due to its large scale, variety of topics, and the availability of aligned source articles and reference summaries.

The summarization task can be formulated as follows. Given a source document represented by a sequence of words $D = w1, w2, \ldots, wn$, the goal is to generate a concise summary $S = s1, s2, \ldots, sm$, where $m \ll n$. The summary should capture the main ideas and information present in the source document.

Mathematically, the summarization task involves finding the optimal summary S∗ that maximizes a predefined objective function:

$$S^* = \arg\max_S Score\,(S, D)$$

Here, $Score(S, D)$ is a scoring function that quantifies the quality of the generated summary S given the source document D. Various metrics, such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation), can be used to evaluate the effectiveness of the generated summaries.

3.2 Data Generation

This step aims to harness the generative capabilities of ChatGPT for creating a dataset with predicted labels. It involves presenting task-specific input prompts, such as providing a set of [text] to ChatGPT, and receiving a set of pairs in return, denoted as [text] : [class], thus forming a pseudo dataset infused with valuable task-specific knowledge.

This article selects a portion of gold results from the CNN training dataset as part of the prompt, instructing ChatGPT to expand the content based on a summarization result. This process is employed to generate training data pairs. For instance, using the following content for news rewriting:

"Please write a news article based on the following news summary. ‘NEW: President Bush says he and first lady are deeply saddened by the tragedy. Mine Safety and Health Administration chief: We've run out of options. The six men have been trapped underground since August 6. Seven bore holes drilled into the mountain have found no signs of life.’"

The generated result is then used as input, while the gold result from the prompt serves as the output, thereby compiling the training dataset.

3.3 Lightweight Small Model Design

With the pseudo dataset generated as described earlier, a Tiny Classical Model is trained for a particular natural language (NLP) task. The Tiny Classical Model, in this context, is significantly smaller in scale compared to ChatGPT, encompassing models like T5, Bart, and others.

This article chooses T5-small and Bart-base as the small models for the study.

3.3.1 T5-small

T5 (Text-to-Text Transfer Transformer) is a natural language processing model developed by Google. T5 adopts the Transformer architecture, with the design philosophy of treating all NLP tasks as text-to-text problems. The model’s input and output are both in the form of text, making T5 highly versatile and adaptable to various natural language processing tasks, such as text classification, named entity recognition, and summarization.

The core idea behind T5 is to transform diverse NLP tasks into a common text generation problem. The training data for the model includes input text and corresponding target text, with different tasks trained by simply altering the target text. This consistent text input-output framework simplifies the design and application of the model.

The training process of T5 involves extensive pre-training and fine-tuning using large corpora. The pre-training task typically involves filling in missing text fragments, allowing the model to learn language structure and semantics. The fine-tuning phase adjusts the model through supervised learning on specific tasks, enabling it to adapt to application domains.

Since its release, the T5 model has become a significant reference in the field of natural language processing,

article	highlights
LONDON, England (Reuters) -- Harry Potter star Daniel Radcliffe gains access to a reported £20 million ($41.1 million) fortune as he turns 18 on Monday, but he insists the money won't cast a spell on him. Daniel Radcliffe as Harry Potter in "Harry Potter and the Order of the Phoenix" To the disappointment of gossip columnists around the world···	Harry Potter star Daniel Radcliffe gets £20M fortune as he turns 18 Monday . Young actor says he has no plans to fritter his cash away . Radcliffe's earnings from first five Potter films have been held in trust fund .
BOLINGBROOK, Illinois (CNN) -- The disappearance of a suburban Chicago police sergeant's wife is now being treated as a potential homicide, and her husband is a suspect, authorities said Friday. Stacy Peterson, 23, has been missing from her suburban Chicago home since October 28. In another development, a judge signed an order to exhume the body of Drew Peterson's third wife···	NEW: Judge signs order to exhume the body of Drew Peterson's third wife . Peterson has said he believed his fourth wife left him for another man . Police: Case shifts from a missing persons search to a potential homicide . Friends and family: Stacy Peterson expressed concerns about her husband .

Figure *2.* CNN Dataset Samples

providing robust performance and versatility for a variety of text-processing tasks.

3.3.2 Bart-based

Bart (BART: Bidirectional and Auto-Regressive Transformers) is a sequence-to-sequence model developed by Facebook AI Research for natural language processing tasks. Unlike traditional natural language processing models, Bart adopts the Transformer architecture and introduces improvements based on it.

What sets Bart apart is its use of bidirectional and auto-regressive training methods. This combination allows Bart to consider contextual information when generating text while retaining some advantages of auto-regressive models.

The training process of Bart primarily consists of two stages: pre-training and fine-tuning. In the pre-training phase, the model engages in self-supervised learning by predicting masked portions of input text using a large-scale corpus. The fine-tuning stage involves adjusting the model using labeled data specific to a given task, enabling it to adapt to a particular application domain.

Bart demonstrates outstanding performance in various natural language processing tasks, including text summarization, text generation, and machine translation. Its flexibility and efficiency make it a powerful tool for handling sequence-to-sequence tasks, contributing significantly to the field of natural language processing.

3.4 Training

Ultimately, the adeptly trained Tiny Classical Model engages in proficient and resource-efficient online inference for the targeted task. For instance, given an input [text], it outputs the corresponding [class]. Throughout this entire process, human annotations are entirely omitted, presenting a scenario akin to a completely zero-shot inference.

4 Experiment

In this chapter, we will provide a detailed explanation of the training methodology using data generated by GPT, focusing specifically on the summarization task. We will compare and analyze the results obtained from training with GPT-generated data with those obtained from training using CNN data.

Table 1: Dataset Count

Dataset	Training num	Testing num
CNN	39626	1900
GPT Generated	23797	/

4.1 Datasets

The CNN dataset is a collection of news data released by CNN (Cable News Network), designed for natural language processing and machine learning tasks. This dataset includes news articles, headlines, and associated metadata, covering a wide range of topics and domains. Due to the extensive size of the dataset and to conserve computational resources, we only select data where the article length is less than 2048 characters. Correspondingly, the same processing has been applied to the ChatGPT-generated data.

After filtering, the CNN dataset has a total of 39,626 training samples and 1,900 test samples. The GPT-generated dataset comprises 23,797 samples.

4.2 Metrics

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used for automatic text summarization evaluation. Three common variants are ROUGE-1, ROUGE-2, and ROUGE-L.

Table 2: Experiment Results

Model	Training data	Test data	Rouge 1	Rouge 2	Rouge L	Avg gen len
T5-small	CNN	CNN test	43.3346	21.5843	31.7666	61.3116
Bart-base	CNN	CNN test	45.0874	23.1264	32.7113	64.5447
T5-small	GPT generated	CNN test	43.0601	21.4929	31.6732	61.2868
Bart-base	GPT generated	CNN test	44.7815	22.8785	32.3875	63.1616
ChatGPT	/	CNN test	37.2353	13.6557	23.2425	92.4464

4.2.1 ROUGE-1

ROUGE-1 measures the overlap of unigrams (single words). Let $N_{overlap-1}$ be the number of overlapping unigrams, and $N_{reference-1}$ be the total number of unigrams in the reference. The ROUGE-1 score is given by:

$$ROUGE-1 = \frac{N_{overlap-1}}{N_{reference-1}}$$

4.2.2 ROUGE-2

ROUGE-2 bigrams (pairs of consecutive words). Let $N_{overlap-2}$ be the number of overlapping bigrams, and $N_{reference-2}$ be the total number of bigrams in the reference. The ROUGE-2 score is given by:

$$ROUGE-2 = \frac{N_{overlap-2}}{N_{reference-2}}$$

4.2.3 ROUGE-L

ROUGE-L measures the longest common subsequence (LCS) between the reference and generated summaries. Let NLCS be the length of the LCS, and $N_{reference-L}$ be the total number of words in the reference. The ROUGE-L score is given by:

$$ROUGE-L = \frac{N_{LCS}}{N_{reference-L}}$$

Here, the LCS represents the length of the longest sequence of words that appears in both the reference and the generated summary.

These ROUGE metrics provide a quantitative evaluation of the quality of machine-generated summaries based on unigram and bigram overlaps, as well as the longest common subsequence.

4.3 Experiment Results

We conducted two sets of control experiments and one set of ChatGPT zero-shot experiments separately. From the experimental results, it can be seen that models trained on T5-small and Bart-base, using GPT-generated dataset, exhibit performance very close to those trained on CNN dataset.Considering that CNN dataset is based on high quality data collected from CNN Daily, while the GPT-generated dataset is simply generated by non-professional language data collectors using Chat-GPT, it indicates that using GPT to generate training data for small models is indeed feasible.

The T5-small model and Bart-base model outperformed the results of ChatGPT Zero-shot in the experimental outcomes. We speculate that this may be due to the average length of the generated results by ChatGPT being longer than the average length of results generated by other models, causing a lower Rouge similarity score during calculation.

The performance of the Bart-base model is better than that of T5-small. This might be attributed to the inherent characteristics of the Bart model, allowing it to capture contextual information more effectively, resulting in slightly superior performance compared to T5-small.

5 Conclusion

This study explores optimizing lightweight small models by generating training data with ChatGPT. The experiments indicate that small models trained on corpora generated by ChatGPT perform very closely to those trained on professionally annotated datasets. This suggests that the quality of language data generated by ChatGPT is comparable to that of professional training datasets. By training small models with corpora generated through ChatGPT, it becomes possible to rapidly acquire high-quality training data. This not only ensures model quality but also enhances the speed of inference.

Acknowledgments

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

Funding

Not applicable.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Conflict of Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Author Contributions

Not applicable.

About the Authors

DING, Rui

Male, Master Student at University of San Francisco Bay University, California, USA.

ZHU, Elly yijun

Female, Independent Researcher, California, USA.

ZHAO, Chao

Female, Master Student at Georgia Institute of Technology, Georgia, USA.

YANG, Haoyu

Male, Master Student at Georgia Institute of Technology, Georgia, USA.

LI, Jing

Female, Independent Researcher, California, USA.

WU, Yue

Male, Independent Researcher, California, USA.

References

[1] Logan, R. L., IV, Balažević, I., & Wallace, E. et al. (2021). Cutting down on prompts and parameters: Simple few-shot learning with language models.

[2] Brown, T. B., Mann, B., & Ryder, N. et al. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

[3] Dai, W., Tao, J., & Yan, X. et al. (2023). Addressing unintended bias in toxicity detection: An LSTM and attention-based approach. In 2023 5th International Conference on Artificial Intelligence and Computer Applications (ICAICA), 375–379.

[4] Devlin, J., Chang, M.-W., & Lee, K. et al. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[5] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 1189–1232.

[6] Hermann, K. M., Kociský, T., & Grefenstette, E. et al. (2015). Teaching machines to read and comprehend. In NIPS, 1693–1701.

[7] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[8] Lewis, M., Liu, Y., & Goyal, N. et al. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language processing. arXiv preprint arXiv:1910.13461.

[9] Li, S., Kou, P., & Ma, M. et al. (2024). Application of semi-supervised learning in image classification: Research on fusion of labeled and unlabeled data. IEEE Access, 12, 27331–27343.

[10] Liu, Y., Yang, H., & Wu, C. (2023). Unveiling patterns: A study on semi-supervised classification of strip surface defects. IEEE Access, 11, 119933–119946.

[11] Liu, T., Xu, C., & Qiao, Y. et al. (2024). News recommendation with attention mechanism. Journal of Industrial Engineering and Applied Science, 2(1), 21–26.

[12] Su, J., Jiang, C., & Jin, X. et al. (2024). Large language models for forecasting and anomaly detection: A systematic literature review. arXiv preprint arXiv:2402.10350.

[13] Mishra, S., Khashabi, D., & Baral, C. et al. (2022). Cross-task generalization via natural language crowdsourcing instructions.

[14] Miyato, T., Maeda, S., & Koyama, M. et al. (2018). Virtual adversarial training: A regularization method for supervised and semi-supervised learning.

[15] Perez, E., Kiela, D., & Cho, K. (2021). True few-shot learning with language models. In Advances in Neural Information Processing Systems, 34, 11054–11070.

[16] Puri, R., & Catanzaro, B. (2019). Zero-shot text classification with generative language models.

[17] Raffel, C., Shazeer, N., & Roberts, A. et al. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.

[18] Sanh, V., Webson, A., & Raffel, C. et al. (2022). Multitask prompted training enables zero-shot task generalization.

[19] Schick, T., & Schütze, H. (2021). Few-shot text generation with pattern-exploiting training.

[20] See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1073–1083.

[21] Tam, D., Menon, R. R., & Bansal, M. et al. (2021). Improving and simplifying pattern exploiting training. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 4980–4991.

[22] Xu, H., Chen, Y., & Du, Y. et al. (2022). ZeroPrompt: Scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization.

[23] Ye, Q., Lin, B. Y., & Ren, X. (2021). CrossFit: A few-shot learning challenge for cross-task generalization in NLP.

[24] Yin, W., Hay, J., & Roth, D. (2019). Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3914–3923.