VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

NGO DUC HUY

TITLE

SUMMER INTERNSHIP REPORT

Major: Computer Science

Supervisor: Dr. Ta Viet Cuong

HANOI - 2024

VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

INTERNSHIP REPORT

TITLE

Student: Ngo Duc Huy
Student ID: 21020046
Class: QH-2021-I/CQ-I-CS2
Supervisor: Dr. Ta Viet Cuong

HANOI - 2024

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Ta Viet Cuong, for his guidance, support, and encouragement throughout the internship. His valuable advices and feedbacks have helped me a lot in completing the research and experiments.

I would also like to the Faculty of Information Technology, University of Engineering and Technology, VNU, and HMI lab for providing me with the opportunity to participate in the summer internship program.

Without the support, I would not have been able to complete the internship and this report.

Abstract

Abstract: This is the report on the research and experiments of modifying and evaluating Large Language Models (LLMs) during the summer internship at the HMI Lab, Faculty of Information Technology, University of Engineering and Technology, VNU, under the guidance of Dr. Ta Viet Cuong.

Keywords: *Keyword*

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Large Language Models and Tiny Language Models

Large Language Models (LLMs) are machine learning models that are capable of processing and understanding natural language through learning from a large amount of text data. They can predict, generate text, answer questions, translate, and perform many other complex language tasks. Thanks to the computational power and large-scale data, these models have made significant breakthroughs in Natural Language Processing (NLP).

LLMs can come in various sizes, from a few millions to billions or trillions of parameters. The larger the model, the better the performance, but also the higher the computational cost. For example, GPT-4, one of the largest LLMs, has 1.8 trillion parameters, which requires a tremendous amount of memory and computational resources to train and deploy.

Therefore, one of the challenges when using LLMs is the high computational cost and resource requirements. To address this issue, Tiny Language Models are introduced, which are smaller versions of LLMs that could be aqquired by creating from scratch or through processes such as knowledge distillation, model finetuning, and other training techniques. These models still retain the core language capabilities like other larger model but with lower resource requirements, faster query processing speed, at a small exchanched cost of performance.

The research and use of tiny language models is a new and relatively important research direction in the field of NLP, helping to optimize the performance and cost

of NLP applications in practice, or in training step, it may reduce the amount of data or energy required, which is beneficial for the environment. This could be the way for small organizations or individuals lacking resources to enter the field of language models research and development.

1.2 Research Objectives

This report presents the research and experiments on a tiny language model created by modifying PhoGPT, a state of the art languege models for Vietnamese. The process including explore the process of modifying the model, evaluating the performance of the model in specific evaluations. In overall, the report aims to answer the following questions:

- Can we modify a LLM like PhoGPT to create a smaller version with fewer parameters and lower resource requirements?
- What is the tradeoff between the performance and the resource requirements of the SLM compared to the original LLM?

Chapter 2

Related Works

2.1 PhoGPT

Model	All truthful questions	Vietnam-specific
PhoGPT-4B-Chat	<u>41.7 (83 / 199)</u>	**43.5 (64 / 147)**
GPT-4-0125-preview	**44.7 (89 / 199)**	39.5 (58 / 147)
GPT-3.5-turbo	29.1 (58 / 199)	22.4 (33 / 147)
Gemini Pro 1.0	39.7 (79 / 199)	34.7 (51 / 147)
Vistral-7B-Chat	41.2 (82 / 199)	<u>42.9 (63 / 147)</u>
Sailor-7B-Chat	28.6 (57 / 199)	27.9 (41 / 147)
Sailor-4B-Chat	15.6 (31 / 199)	14.3 (21 / 147)
SeaLLM-7B-v2	20.6 (41 / 199)	13.6 (20 / 147)
VBD-Llama2-7B-50B-Chat	15.6 (31 / 199)	10.9 (16 / 147)
Vinallama-7B-Chat	11.1 (22 / 199)	8.2 (12 / 147)
Gemma-7B-it	8.0 (16 / 199)	6.1 (9 / 147)

Table 2.1: *PhoGPT-4B-Chat and other models' performance on the truthful questions and Vietnam-specific questions tasks. The best model is listed in bold and second-best is listed in underlined.*

PhoGPT [1] is a open-source state-of-the-art Transformer decoder-based model series for Vietnamese, including the base pre-trained monolingual model PhoGPT-4B and its chat variant, PhoGPT-4B-Chat. The base model, PhoGPT-4B, with exactly 3.7B parameters, is pre-trained from scratch on a Vietnamese corpus of 102B tokens, with

an 8192 context length, employing a vocabulary of 20480 token types. The chat variant, PhoGPT-4B-Chat, is the modeling output obtained by fine-tuning PhoGPT-4B on a dataset of 70K instructional prompts and their responses, along with an additional 290K conversations.

The PhoGPT series has achieved state-of-the-art results on various Vietnamese NLP tasks in questions form as showed in 2.1

2.2 Tiny Language Models

The work of Yehui Tang et al. [2] introduces a series of experiments on Tiny Language Models based on Pangu, a large-scale language model on Chinese and English. The authors propose a method to compress Pangu into Pangu-π-1B-pro and Pangu-π-1.5B-pro with 1B and 1.5B parameters, respectively. The experiments show that the small models can achieve competitive performance with a much smaller number of parameters compared to the original model as showed in Table 2.2 and Figure 2.1

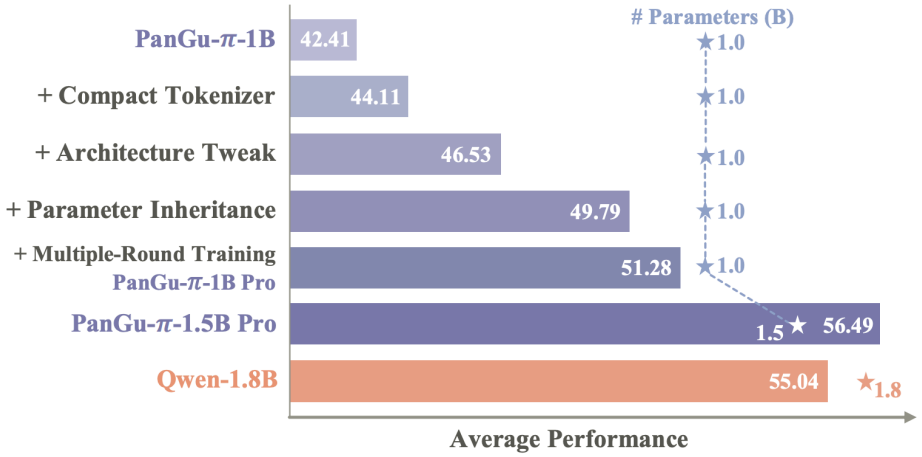

Figure 2.1: *Pangu-π-1B-pro and Pangu-π-1.5B-pro performance compared to other small model.*

The authors propose many methods for creating the models, from the model architecture, parameter initialization, to the optimized training strategies. Firstly, for the model architecture, there are two main components: the tokenizer and the model itself:

- As for the tokenizer, the authors after conducting experiments with different vocabulary sizes, they came to the conclusion that over 50% vocabularies may be redundant as they cater to less than 3% of the corpus. Therefore, they propose to

Models	Examination				Knowledge	Reasoning		Understanding			Average
	C-Eval	CMMLU	MMLU	AGI-Eval	BoolQ	AX-b	PIQA	EPRSTMT	XSum	C3	
MobileLLaMA-1.4B	23.93	25.10	25.05	18.53	58.75	45.20	71.27	46.25	18.19	37.42	36.97
Sheared-LLaMA-1.3B	24.28	25.10	25.77	18.01	62.39	43.57	72.91	46.25	16.44	35.45	37.02
TinyLLaMA-1.1B	27.85	24.64	25.75	18.54	56.06	45.47	70.62	46.25	20.15	36.71	37.20
MobileLLaMA-2.7B	23.53	25.55	26.63	18.43	54.74	55.80	72.85	46.25	16.96	36.11	37.69
Chinese-LLaMA2-1.3B	28.70	24.78	24.55	19.40	56.79	47.46	56.91	72.50	8.90	43.12	38.31
RWKV-5-1.5B	25.92	25.14	25.66	19.01	62.29	54.05	71.22	46.25	<u>20.67</u>	49.15	39.94
Phi-1.3B	27.78	25.85	44.32	23.42	<u>73.52</u>	44.20	76.99	50.00	14.90	38.96	41.99
PanGu-π-1B	36.85	35.90	35.96	30.77	58.44	43.48	61.92	55.62	15.92	49.21	42.41
Open-LLaMA-3B	27.50	25.42	27.09	20.68	60.58	52.72	<u>77.09</u>	82.50	19.75	43.23	43.66
Phi2-2.7B	31.86	32.18	**58.49**	28.51	**77.40**	43.57	**78.89**	46.25	13.66	40.11	45.09
PanGu-π-1B Pro (**Ours**)	46.50	46.56	50.38	<u>41.58</u>	63.43	53.99	64.96	74.38	18.40	52.66	51.28
Qwen-1.8B	**53.60**	**52.12**	46.43	35.83	64.31	**57.79**	73.83	<u>88.12</u>	20.03	<u>58.30</u>	<u>55.04</u>
PanGu-π-1.5B Pro (**Ours**)	<u>52.91</u>	<u>49.51</u>	<u>53.76</u>	**44.42**	63.73	<u>55.93</u>	73.94	**89.38**	**22.23**	**59.56**	**56.49**

Table 2.2: *Comparison with SOTA open-source tiny language models. The best model is listed in bold and second-best is listed in underlined.*

choose the tokenizer that cover 90% of the corpus vocabulary, which is the reduced one from Pangu-π model, resulting in a vocabulary size of 48K instead of 100K tokens.

- And for the model architecture tweaking, the authors explored the impact of depth, width and the expanding rate of Feed-Forward Networks (FFN) on the performance of a 1B-size language model. They found that the depth of the model has a significant impact on the performance and inference speed, while the width and the expanding rate of FFN have a minor impact.

Secondly, when working with parameter initialization methods like random initialization and parameter inheritance, the authors found that the latter method can help the model perform better by preserving the knowledge of the pre-trained model and reducing the training time.

By using both important inter-layer selection and intra-layer parameter selection strategies, the authors can choose which layers and neurons to inherit to the model, keeping the most important elements that contribute to the larger model's performance while vastly reducing the number of parameters.

Finally, the authors propose a series of optimized training strategies after experimenting with batch size and learning rate, in combination with multiple-round training. This results in less hardware consumption while rectifying the problem of data forgetting due to the limitation in capacity of the model.

Chapter 3

Research methodology

In this chapter, the methods used to conduct the research and answer to the 2 research questions mentioned previously will be presented. Firstly, the method for modifying the model architecture will be discussed. And then the performance metrics and the results of the modified model will be presented.

3.1 Model architecture modification

The model architecture modification is done by changing the model's architecture to a smaller one. One of the most common ways to reduce the size of the model is to reduce the number of layers in the model. The layers in the concept of a generative Transformer LLM is referred to as the number of Transformer blocks placed consecutively in the model. The Transformer block is the basic building block of the Transformer model, which consists of a multi-head self-attention mechanism and a feed-forward neural network. The number of Transformer blocks in a model is the most important factor that determines the model's size. Therefore, reducing the number of Transformer blocks in the model will reduce the model's size significantly.

Since the blocks are placed consecutively in the model, i.e., the output of the previous block is the input of the next block, the process happens layer by layer automatically. Therefore layer reduction can be done by simply removing the layers from the model.

3.2 Metrics and performance

The performance of the modified model is evaluated using the perplexity metric. Perplexity is a common metric used to evaluate the performance of a language model. It is calculated as the exponential of the cross-entropy loss of the model on the test dataset. The lower the perplexity, the better the model's performance.

The perplexity is calculated as follows:

$$\text{Perplexity} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log p(x_i|x_{<i})\right) \tag{3.1}$$

where N is the number of tokens in the sequence

$p(x_i|x_{1:i-1})$ is the probability of the i-th token x_i given the preceding tokens $x_{1:i-1}$

The performance of the model is evaluated by measuring the perplexity of the model on the test dataset consisting of input prompts sequences. The process is done by feeding the input sequences to the model and calculating the perplexity of the model on the output sequences. Then the perplexity is averaged over the whole test dataset to get the final perplexity score of the model.

Chapter 4

Experiments and Results

Experiment	Avg. PPL	Increase in PPL	Avg. Runtime	Decrease in Runtime
Baseline	*12.6439*	*0.00%*	*0.6476*	*0.00%*
1 layer - first	23508589055.1685	185928312366.07%		
1 layer - middle	13.2905	5.11%	0.5611	-13.36%
1 layer - last	22.6169	78.88%		
2 layer - first	2361030.4434	18673177.46%		
2 layer - middle	13.8172	9.28%	0.5449	-15.85%
2 layer - last	23.6517	87.06%		
4 layer - first	17938.6214	141775.70%		
4 layer - middle	16.6120	31.38%	0.5097	-21.30%
4 layer - last	29.4387	132.83%		
8 layer - first	4740882052.1188	37495410512.80%		
8 layer - middle	28.1565	122.69%	0.4388	-32.25%
8 layer - last	52.1473	312.43%		

Table 4.1: *Results of Experiments*

The experiments are conducted first by using the original PhoGPT-4B-Chat model as the baseline model. With 24 Transformer blocks or layers, each with 24 attention heads, the model results in roughly 3.7B parameters.

The model is then modified in multiple ways to reduce the model size by removing different numbers and positions of layers from the model. We had experimented with removing 1, 2, 4, 8 layers consecutively, repeating from the first layer to the last layer possible of the model. The modified model are then evaluated using the perplexity metric on the test prompt dataset. The average perplexity is calculated as metrics to

evaluate the performance of the model.

The experiments are conducted on a single NVIDIA RTX 3090Ti GPU with 24GB of VRAM, given the following results in Table 4.1.

As can be seen from the table, the baseline model PhoGPT-4B-Chat has a perplexity of 12.6439 on the test dataset. When removing 1 layer from the model, the perplexity increases to 13.2905 (about 5%) as average, to tremendously large when removing the critical layers. The results pattern continues as more layers are removed from the model. The perplexity increases as the number of layers removed increases, the first and last layers have much more significant impact on the model's performance compared to the middle layers.

The runtime or inference speed of the model is also measured during the experiments. The runtime of the model is measured by calculating the average time taken to generate the output sequences for the test dataset. As the model size decreases, the runtime of the model also decreases. The runtime of the model is inversely proportional to the model size, as the trade-off between model size or performance and inference speed is a common problem in the battlefield of LLMs.

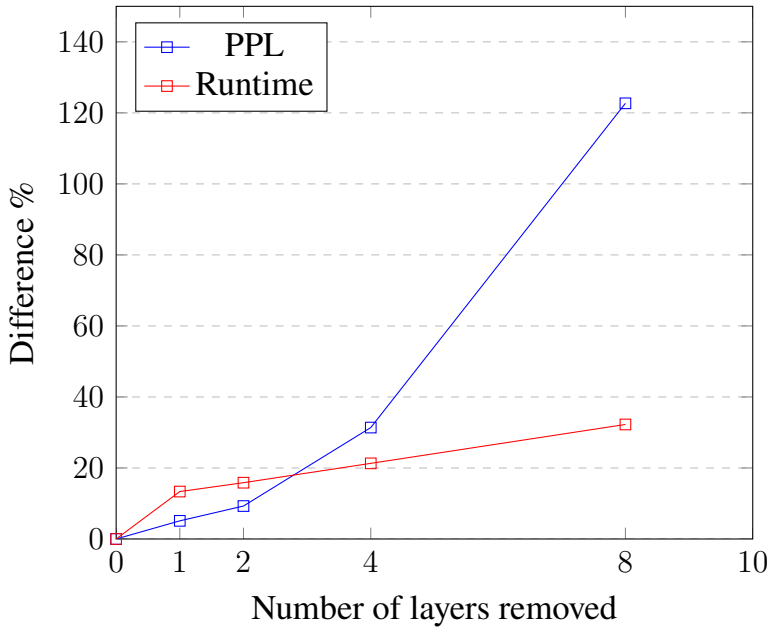

Figure 4.1: *Model's Perplexity and Runtime results with layers removed*

Conclusion

In this internship report, we have presented the work on the modification of the PhoGPT-4B-Chat model to reduce the model size while not significantly affecting the model's performance. The experiments are conducted by removing different numbers and positions of layers from the model and evaluating the model's performance using the perplexity metric on the test prompt dataset. The results show that the trade-off between model size and performance, as well as the inference speed. The results also show one important feature of the model: the first and last layers have much more significant impact on the model's performance compared to the middle layers.

References

[1] D. Q. Nguyen, L. T. Nguyen, C. Tran, D. N. Nguyen, D. Phung, and H. Bui, "Phogpt: Generative pre-training for vietnamese," 2024. [Online]. Available: https://arxiv.org/abs/2311.02945

[2] Y. Tang, F. Liu, Y. Ni, Y. Tian, Z. Bai, Y.-Q. Hu, S. Liu, S. Jui, K. Han, and Y. Wang, "Rethinking optimization and architecture for tiny language models," 2024. [Online]. Available: https://arxiv.org/abs/2402.02791

Supervisor's comment:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Mark: _____ **In words:** _____

Hanoi, _____/_____/2024

Supervisor

(Signature and full name)