IT-Tuning : Parameter Efficient Information Token Tuning for Language Model

Jungu Kim, Hyeoncheol Kim*
Department of of Computer Science and Engineering, Korea University
{antonio97k, harrykim}@korea.ac.kr

Abstract

Recently, language models have demonstrated exceptional performance compared to their predecessors. In this context, attention mechanisms and pre-training significantly contribute to the enhanced performance of modern language models. Additionally, a continuously increasing number of parameters plays a crucial role in these advancements. However, an increase in the number of parameters significantly increases the GPU memory and training time required during fine-tuning of language models, this makes fine-tuning infeasible in environments with limited computing resources. Furthermore, after fine-tuning, the storage space required for deployment increases proportionally with the number of tasks, making it challenging to deploy devices with limited storage capacities. In this study, we propose IT-Tuning, a Parameter Efficient Fine-Tuning method that introduces a new concept called information tokens to address these issues.

1 Introduction

Since the introduction of Transformer(Vaswani et al., 2017) and BERT(Devlin et al., 2019), recent Transformer based pre-trained language models have achieved unprecedented performance, coinciding with an increase in the number of parameters.(Kaplan et al., 2020; Brown et al., 2020; Chowdhery et al., 2024; Touvron et al., 2023) Consequently, research on various applications, such as sentiment analysis, question answering, sentence classification, summarization, and machine translation, has been actively conducted. Although pretrained language models can be utilized in various tasks using only prompts without fine-tuning, as demonstrated by approaches such as chain-of-thought prompting(Wei et al., 2022) or in-context learning(Dong et al., 2022), fine-tuning often leads to better performance. However, recently introduced language models have billions to tens of billions of parameters(Zhao et al., 2023), resulting in increased GPU memory and extended training time requirements during fine-tuning. In addition, deploying these models after fine-tuning across various tasks requires significant storage space proportional to the size of the model and the number of tasks, which poses a challenge.

In this context, we introduce a new concept called "information token" to address this issue. The information token attends to all tokens within the input sentence during the attention mechanism process, selectively condensing the information of the sentence according to the task and delivering it to the input and output sentences, enabling efficient fine-tuning. Based on these information tokens, we propose a new Parameter Efficient Fine-Tuning (PEFT) method called IT-Tuning, the contributions of which are as follows:

1. We introduce a new concept called information tokens to efficiently fine-tune language models and demonstrate experimental methods to adjust them more efficiently within the model.

2. Using only 0.04% of the total parameters, which is five times less than LoRA(Hu et al., 2021), we surpass LoRA and full fine-tuning in the General Language Understanding Evaluation (GLUE) benchmark(Wang et al., 2018), demonstrating efficiency in Natural Language Understanding (NLU) tasks.

3. We have achieved performance surpassing that of full fine-tuning, Prefix Tuning(Li and Liang, 2021), and LoRA using only 0.09% of the total number of parameters in Natural Language Generation (NLG) tasks, as demonstrated through experiments on the End-to-End Natural Language Generation Challenge (E2E NLG Challenge) dataset(Dušek et al., 2020).

Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024), pages 58–68
August 15, 2024 ©2024 Association for Computational Linguistics

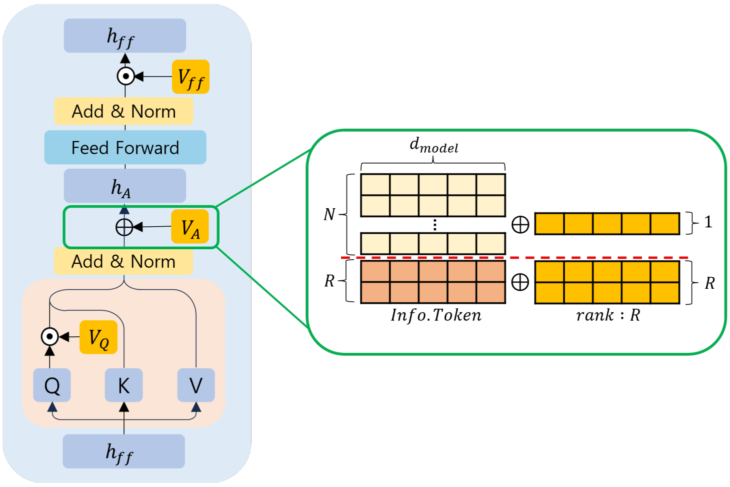


Figure 1: This is the overall architecture of IT-Tuning. As shown in the figure, we perform shifting or scaling after the query vector, the output of the attention, and the feed-forward layer. Additionally, the yellow and red vectors within the green box respectively denote the hidden states of all tokens of the input sentence and the information tokens. N and R(rank) represent the length of the input sentence and the number of additional information tokens. Inside the model, we scale or shift all tokens of the input sentence excluding the information tokens using a single vector to assist the role of information tokens described in Section 3.2. For information tokens, we adjusted each using an equal number of vectors.

2 Related Works

Recently, extensive research has been conducted on PEFT, achieving a performance equivalent to that of full fine-tuning using only 0.1% of the total model parameters.(Hu et al., 2021; Li and Liang, 2021; Liu et al., 2022a; Yang et al., 2023) This approach significantly reduces the size of the GPU memory and training time required. Furthermore, because PEFT requires only a few additional parameters to be stored when a single model is used for various tasks, it saves storage space. In this section, we describe the research on PEFT conducted to date.

Adapter It is a method of inserting multiple adapter modules (MLP modules) into the layers of a language model. Research has continued extensively after the Adapter(Houlsby et al., 2019), and recent studies such as AdapterBias(Fu et al., 2022) and AdaMix(Wang et al., 2022) have significantly improved performance by modifying the structure of the adapter module or proposing the Mixture-of-Adaptation method. Furthermore, adapter-based PEFT methods can adjust the number of parameters used in training by altering the number of parameters in the adapter module, making them applicable to both NLU and NLG tasks.

Trainable Prompt This method involves adding trainable tokens to sentences, with prominent examples of P-Tuning v1 and v2(Liu et al., 2022b, 2021), Prefix Tuning(Li and Liang, 2021), and Prompt Tuning(Lester et al., 2021) . These studies aimed to address the performance gap observed when using sentence-form prompts (discrete prompts) compared with fine-tuned language models by incorporating trainable tokens through fine-tuning rather than nontrainable sentence-form prompts. These studies have demonstrated results achieving equivalent performance to full fine-tuning in NLG(Li and Liang, 2021) tasks and NLU(Liu et al., 2022b, 2021; Lester et al., 2021) tasks.

Low-Rank PEFT methodologies based on low-rank, such as LoRA(Hu et al., 2021) and HiWi(Liao et al., 2023), are gaining attention owing to their robust performance and capability to mitigate the issue of increased inference times associated with additional parameters. However, one of the significant advantages of PEFT is its ability to switch tasks in a multitasking environment, which is compromised by combining model parameters and low-rank parameters to mitigate the increase in inference time. Additionally, there are studies such as (IA)^3(Liu et al., 2022a) that achieve better performance with fewer parameters than LoRA in some tasks. Nonetheless, LoRA has proven to be a robust method that is effective in both NLU and NLG tasks. Therefore, we conducted experiments using the LoRA as the baseline.

Direct Update This method directly adjusts the hidden states of the model using the added vectors.

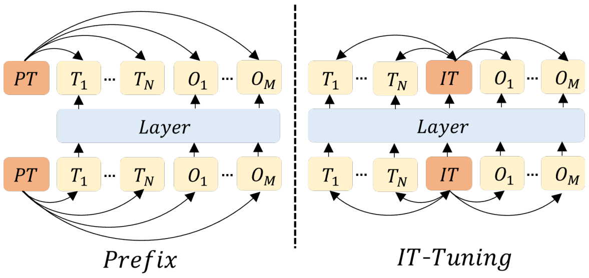


Figure 2: This illustrates the difference in attention mechanisms between Prefix Tuning(Li and Liang, 2021) and IT-Tuning. PT represents additional tokens in Prefix Tuning, IT represents information tokens. The arrows indicate which tokens attend to which tokens.

For PASTA(Yang et al., 2023), only the hidden states of special tokens, such as [CLS] and [SEP], were updated at each layer of the model, resulting in an additional parameter number of 0.02% (0.07M) based on RoBERTa-large(Liu et al., 2019), while achieving a performance equivalent to LoRA on the GLUE dataset. Furthermore, (IA)[3](Liu et al., 2022a) scales only the key and value of the attention operation and the internal parameters of the feed-forward network in each layer, surpassing LoRA in limited-data settings. However, although these approaches are structurally efficient, their inability to adjust the number of trainable parameters renders them unsuitable for tasks requiring more parameters than NLU, such as NLG. IT-Tuning exhibits structural efficiency akin to that observed in two referenced studies. However, by addressing the limitations associated with parameter adjustments through the introduction of information tokens, it also demonstrates applicability to NLG tasks .

3 IT-Tuning

3.1 Model Architecture

Figure 1 illustrates the model structure of IT-Tuning. Our IT-Tuning employs a method that updates the selected tokens using additional parameters. We refer to these selected tokens as information tokens and introduce this concept in Section 3.2. Furthermore, we introduce an efficient structure to enable the efficient update of information tokens within the model in Section 3.3. The operational process of IT-Tuning within the model can be mathematically represented as follows:

$$\begin{aligned} h_Q &= [h_Q^{input} \odot V_Q^1; h_Q^{it} \odot V_Q^2] \\ h_A &= [h_A^{input} \oplus V_A^1; h_A^{it} \oplus V_A^2] \\ h_{ff} &= [h_{ff}^{input} \odot V_{ff}^1; h_{ff}^{it} \odot V_{ff}^2] \end{aligned} \tag{1}$$

In Equation 1, $h_Q, h_A, h_{ff} \in \mathbb{R}^{d_{model} \times (N+R)}$ represent the hidden state of the query vector and output of the attention and feedforward layer containing information tokens, respectively. $h_Q^{input}, h_A^{input}, h_{ff}^{input} \in \mathbb{R}^{d_{model} \times N}$ represent the inputs of the model, excluding the information tokens, and $h_Q^{it}, h_A^{it}, h_{ff}^{it} \in \mathbb{R}^{d_{model} \times R}$ represent the hidden states of the information tokens $V_Q^1, V_A^1, V_{ff}^1 \in \mathbb{R}^{d_{model}}$ are vectors added to efficiently fine-tune the input tokens, excluding the information tokens; $V_Q^2, V_A^2, V_{ff}^2 \in \mathbb{R}^{d_{model} \times R}$ represent vectors added to efficiently fine-tune the information tokens.

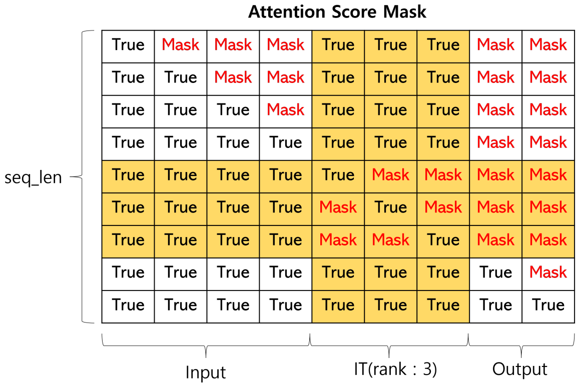
During our experimentation process, we examined the magnitude of backpropagated gradients through learning for scaling vectors V_Q, V_{ff}, as well as for shifting vector V_A. We discovered that the magnitude of gradients for the vectors for scaling, V_Q, V_{ff}, was significantly smaller than the gradients for the vector for shifting, V_A, across all layers. To address this issue, similar to LoRA+(Hayou et al., 2024), we varied the learning rate applied to each vector. To achieve this, we introduced a new parameter, $\alpha \geq 1$, where the magnitude of the learning rate applied to V_Q, V_{ff} is determined based on the value of α. This can be represented mathematically as follows:.

$$\begin{aligned} V_Q &= V_Q - \alpha\eta \times G_{V_Q} \\ V_{ff} &= V_{ff} - \alpha\eta \times G_{V_{ff}} \end{aligned} \tag{2}$$

In equation 2, η represents the learning rate, and G denotes the gradients for each vector.

3.2 Information Token

In this study, we introduce the concept of Information Tokens. Information Tokens are tokens selected or added within a sentence for efficient fine-tuning and are updated by individual vectors. Figure 2 illustrates the differences between the Prefix Tuning(Li and Liang, 2021) and IT-Tuning. As shown on the left side of Figure 2, Prefix Tuning adjusts the tokens added to the beginning of the input according to the task and updates both the input and output by attending to the tokens added within the input sentence. However, the tokens added at the beginning of the input cannot attend to the tokens


Attention Score Mask
seq_len
Input
IT(rank : 3)
Output

True	Mask	Mask	Mask	True	True	True	Mask	Mask
True	True	Mask	Mask	True	True	True	Mask	Mask
True	True	True	Mask	True	True	True	Mask	Mask
True	True	True	True	True	True	True	Mask	Mask
True	True	True	True	True	Mask	Mask	Mask	Mask
True	True	True	True	Mask	True	Mask	Mask	Mask
True	True	True	True	Mask	Mask	True	Mask	Mask
True	True	True	True	True	True	True	True	Mask
True	True	True	True	True	True	True	True	True

Figure 3: In unidirectional language models like GPT(Radford et al., 2019), we modified the attention score mask to allow each token to attend as shown in Figure 2

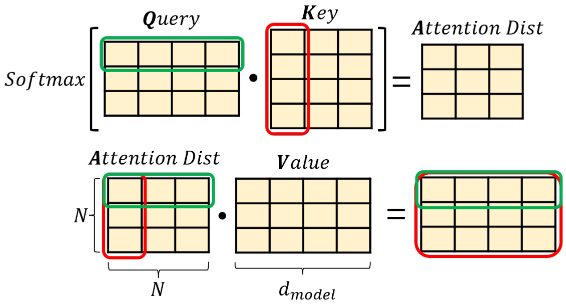


Figure 4: This figure illustrates which hidden states need to be updated for Information Tokens to select and summarize tokens of the input sentence. In the figure, the green box represents hidden states influenced by updating the query vector, while the red box represents hidden states influenced by updating the key vector. For instance, if the key vector is updated as described in (IA)^3(Liu et al., 2022a), as illustrated by the attention distribution in the figure, this adjustment alters the extent to which the tokens of the input sentence pay attention to the Information Tokens.

within the input sentence. However, Information Tokens, as shown on the right side of Figure 2, attend to all tokens within the input sentence during the attention operation process; conversely, all tokens within both the input and output sentences can also attend to Information Tokens. Ultimately, Information Tokens select and summarize information from the input sentence according to the task and convey this summarized information to both the input and output sentences.

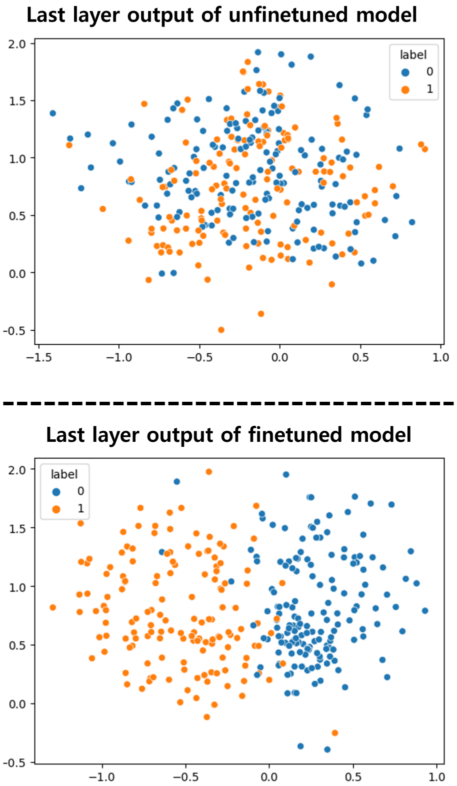
For Information Tokens to fulfill these roles during the attention process, both bidirectional language models (e.g., BERT(Devlin et al., 2019), RoBERTa(Liu et al., 2019)) and unidirectional language models (e.g., GPT-2(Radford et al., 2019)) must be capable of performing bidirectional attention. Therefore, we modified the attention mask in the unidirectional language model, as shown in Figure 3. In addition, when multiple Information Tokens are added, as shown in Figure 3, each Information Token is prevented from attending to another. This allows each Information Token to attend to the input tokens rather than to each other during the learning process. Thus, we enable Information Tokens to interact with all tokens within a sentence in unidirectional language models, ultimately contributing to the prediction of the next token.

Furthermore, unlike in (IA)^3(Liu et al., 2022a), in which update the key vectors, our method involves updating the query vectors. As shown in Figure 4, when updating the key vector of the i-th token, this update adjusts how other tokens select the i-th token, rather than how the i-th token selects other tokens. However, our purpose was to enable the Information Tokens to selectively attend to information in the input sentence according to the task. Therefore, by updating the query vectors of the Information Tokens, as shown in Figure 4, we can enable the Information Tokens to selectively encapsulate important tokens within the input.

3.3 Efficient Structure

In this section, we investigate shifting (addition) and scaling (multiplication) vector operations to adjust the Information Tokens more efficiently within the model. Figure 5 shows the dimensionality reduction of the [CLS] token from the last layer of the BERT model before and after full fine-tuning using the RTE dataset within GLUE(Wang et al., 2018) using t-SNE(Van der Maaten and Hinton, 2008). As is evident from this visualization, language models may seem complex, but in reality, we believe it is a simple process of adjusting the model's parameters through fine-tuning to transform the hidden states into a form that is easily classifiable by the classification layers, as shown in Figure 5. However, rather than fine-tuning the model parameters, our goal was to update the hidden states directly to achieve similar effects, as depicted in Figure 5. To achieve this goal, using both scaling and shifting simultaneously, as in SSF(Lian et al., 2022), is more efficient for transforming vectors compared to the structures of (IA)^3 or PASTA(Yang et al., 2023), which utilize only shifting or scaling operations.

To validate our idea, we conducted preliminary



Last layer output of unfinetuned model

Last layer output of finetuned model

Figure 5: Both the left and right depict visualizations of the [CLS] token from the 12th(last) layer of the BERT-base model when inputting the RTE dataset. The top represents the state after the BERT model has undergone pretraining only, while the down represents the state after conducting full fine-tuning using the RTE dataset.

experiments on tasks within the GLUE dataset before proceeding with the main experiments. Using the BERT-base model, we experimented with five different learning rates for each combination of operations and selected the best-performing. Experimental results, shown in Figure 6, demonstrate that using a combination of one shifting and two scaling operations yields better performance on both large and small datasets than using only shifting or scaling operations. Therefore, we deviate from the framework of previous studies that used single operations to update and enhance IT-Tuning performance by appropriately combining shifting and scaling, as illustrated in Figure 1.

4 Experiment

In this section, experiments are conducted on both NLU and NLG tasks to demonstrate the effectiveness of IT-Tuning. In NLU tasks, we selected the [CLS] token as the information token, while in NLG tasks, we inserted real words such as "summarize", "transformation", "text", "table", etc., between input and output sentences and selected them as information tokens for experimentation.

4.1 NLU

4.1.1 Dataset

In this experiment, we validate the efficiency of IT-Tuning for NLU tasks using the GLUE benchmark. GLUE(Wang et al., 2018) consists of eight datasets: The Corpus of Linguistic Acceptability (CoLA), Stanford Sentiment Treebank (SST-2), Microsoft Research Paraphrase Corpus (MRPC), Semantic Textual Similarity Benchmark (STS-B), Quora Question Pairs (QQP), MultiNLI (MNLI), Question NLI (QNLI), and Recognizing Textual Entailment (RTE). We conducted experiments using these datasets, excluding the WNLI. We used the RoBERTa-large model(Liu et al., 2019) for experimentation, with both rank and α set to 1. Additionally, although the RoBERTa paper suggests using a model pretrained on the MNLI dataset as the initial model when experimenting with small datasets such as RTE, MRPC, and STS-B, we chose not to employ this approach. This is because fine-tuning a pretrained language model on the MNLI dataset is not parameter efficient.

4.1.2 Result

Table 1 presents the results of the experiments with the GLUE dataset using IT-Tuning applied to RoBERTa-large. In Table 1, we report the performance using ACC for the entire validation dataset of the MNLI (matched and mismatched), Matthew's correlation for the CoLA, Pearson's correlation for the STS-B, and ACC for the remaining datasets. Experimental results show that IT-Tuning achieves a performance comparable to that of full fine-tuning and LoRA(Hu et al., 2021) across the GLUE dataset. Notably, IT-Tuning outperforms both full fine-tuning and LoRA using only 0.04% (0.14M) of the parameters of the entire model on the CoLA, MRPC, and RTE datasets with less than 10K training data, excluding STS-B. Moreover, the average IT-Tuning scores surpassed those of both the full fine-tuning and LoRA. Through these experimental results, we demonstrate the high efficiency of IT-Tuning for NLU tasks.

4.2 NLG

4.2.1 Dataset

To demonstrate the effectiveness of IT-Tuning in both NLU and NLG tasks, we conducted exper-

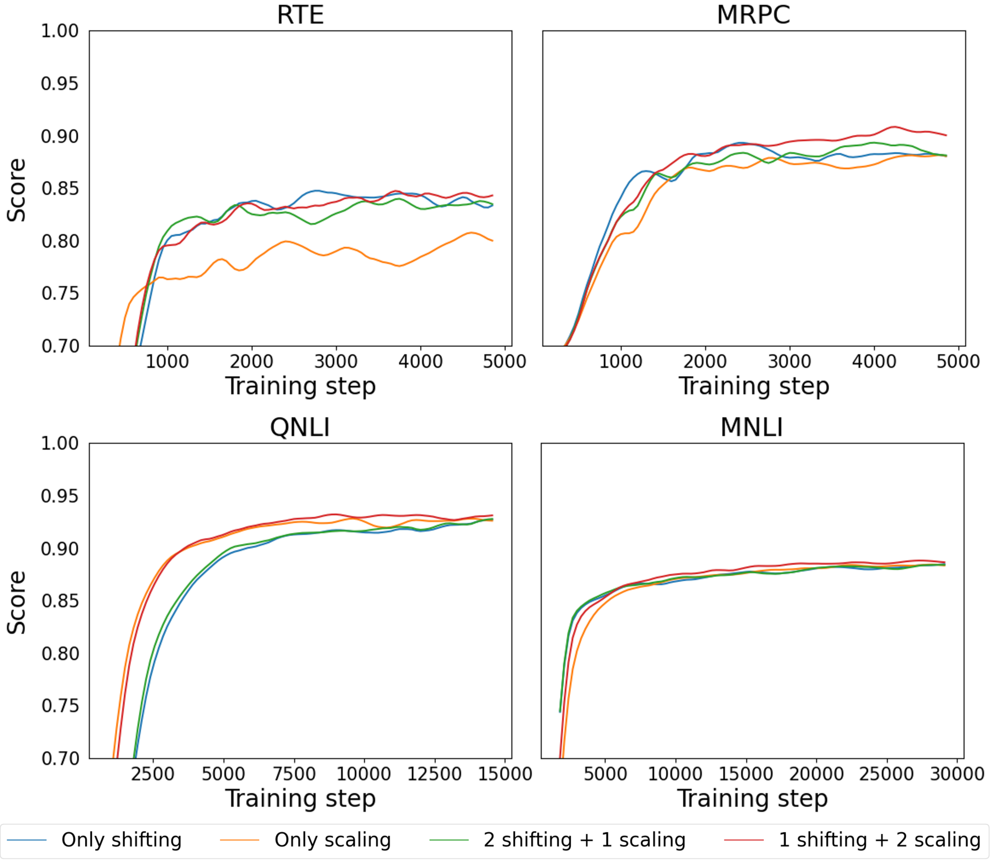


Figure 6: We conducted experiments by updating the hidden states of the query vector, the output of the attention, and the feed-forward layer. In the diagram, "Only shifting" and "Only scaling" represent using only shifting or scaling for all three hidden states, respectively. "2 Shifting + 1 Scaling" applies scaling to the hidden state of the query vector and shifting to the others. "1 Shifting + 2 Scaling" follows the same structure as Figure 1

iments using the E2E NLG Challenge dataset (Dušek et al., 2020). The E2E NLG Challenge dataset comprises approximately 42,000 training instances and 4,600 validation instances for table-to-text evaluation. Each input consisted of slot-value pairs, and the output is a sentence generated based on the input data. We conducted experiments using the GPT2-large model(Radford et al., 2019), with a rank of 4, and an α of 2.

4.2.2 Result

Table 2 presents the results of experiments with the E2E NLG Challenge dataset using IT-Tuning applied to GPT2-large. In Table 2, we evaluated the sentences generated by the model using BLEU(Papineni et al., 2002), NIST(Doddington, 2002), METEOR(Banerjee and Lavie, 2005), ROUGE-L(Lin, 2004), and CIDEr(Vedantam et al., 2015). Experimental results demonstrate the efficiency of our IT-Tuning in both NLU and NLG tasks. Despite using the fewest parameters among all the methods in Table 2, our IT-Tuning has demonstrated astonishing performance surpassing both full fine-tuning and Prefix Tuning(Li and Liang, 2021), as well as LoRA(Hu et al., 2021). Through experiments in NLG tasks, we demonstrated that IT tuning applies not only to NLU but also to NLG. These findings suggest that IT-tuning can serve as a viable alternative to full fine-tuning.

4.3 Ablation Study

4.3.1 Number of Information Token

For our NLU task, we employed one information token, while for NLG tasks, we utilized four information tokens. The ability to adjust the number of information tokens allows us to control the number of parameters used in training, which is one of the significant advantages of IT-Tuning. In this section, we substantiate this aspect. The Table 3 demonstrates the performance variations observed when adjusting the number of information tokens

Model & Method	# Trainable Parameter	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	Avg.
RoB$_{large}$(FT)*	355.0M	68.0	96.4	90.9	92.4	**92.2**	90.2	94.7	86.6	88.92
RoB$_{large}$(Adpt^P)†	3.0M	68.3	96.1	90.2	92.1	91.9	90.2	94.8	83.8	88.42
RoB$_{large}$(Adpt^P)†	0.8M	67.8	96.6	89.7	91.9	91.7	90.5	94.8	80.1	87.88
RoB$_{large}$(Adpt^H)†	6.0M	66.5	96.2	88.7	91.0	92.1	89.9	94.7	83.4	87.81
RoB$_{large}$(Adpt^H)†	0.8M	66.3	96.3	87.7	91.5	91.5	90.3	94.7	72.9	86.40
RoB$_{large}$(LoRA)†	0.8M	68.2	96.2	90.9	**92.6**	91.6	**90.6**	94.9	87.4	89.05
RoB$_{large}$(PASTA)††	0.07M	**69.7**	**96.8**	90.9	91.8	89.9	90.4	**95.1**	86.6	88.90
RoB$_{large}$(ITT)	0.14M	69.5	96.7	**92.2**	91.9	89.7	90.0	94.3	**88.4**	**89.08**

Table 1: RoBERTa-large model performance on GLUE benchmark. In this table, † represents the experimental results of LoRA(Hu et al., 2021), †† indicates the experimental results of PASTA(Yang et al., 2023) **Bold** indicates the best, while underlining indicates the second best.

Model & Method	# Trainable Parameter	BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 L (FT)*	774.03M	68.5	8.78	46.0	69.9	2.45
GPT-2 L (Adpt^L)†	23.00M	68.9	8.70	46.3	71.3	2.49
GPT-2 L (LoRA)†	0.77M	70.4	**8.89**	46.8	72.0	2.47
GPT-2 L (PrefixTuning)††	0.77M	70.3	8.85	46.2	71.7	2.47
GPT-2 L (ITT)	0.69M	**73.4**	8.75	**49.1**	**76.1**	**2.52**

Table 2: GPT2-large model performance on E2E NLG Challenge dataset. In this table, † represents the experimental results of LoRA(Hu et al., 2021), and †† indicates the experimental results of Prefix Tuning(Li and Liang, 2021) Higher is better for all metrics.

in NLG tasks. Notably, the Table 3 reveals a stark BLEU score of 0.07 when employing only one information token. These experimental findings underscore the critical importance of scalability in IT-Tuning. Furthermore, through our experimentation, we observed that as the rank increases, the rate of decrease in training loss accelerates; however, it also becomes easier to encounter overfitting issues.

Model & Method	# Trainable Parameter	BLEU	ROUGE-L
GPT-2 L (ITT)			
- rank : 1	0.27M	0.07	7.65
- rank : 2	0.41M	71.9	73.4
- rank : 4	0.69M	**73.4**	**76.1**
- rank : 8	1.24M	73.0	75.4
- rank : 4 & key	0.69M	72.9	75.1

Table 3: In this table, "rank" denotes the number of information tokens utilized in the experiment, while "key" signifies that key was updated instead of query. Other experimental hyperparameters remain consistent with those outlined in Section 4.2

4.3.2 Using Key instead of Query

As described in the section 3.2, the roles of key and query within attention are different. We updated the query to focus on selectively condensing the information of input sentences, which is a main role of information tokens. However, to further explore the impact of updating key vectors on the performance of IT-Tuning, we conducted experiments in the same environment as described in Section 4.2. In Table 3, experimental results show a slight decrease in performance when using key vectors, but they still demonstrate efficiency. We believe these experimental results highlight the importance not only of how much information tokens pay attention to the other tokens but also of how much other tokens pay attention to the information tokens. Therefore, we believe that combining and updating query and key appropriately to make IT-Tuning more efficient will also be an interesting future work.

5 Conclusion

In this study, we propose information tokens to efficiently learn various tasks such as prediction, classification, and generation by selectively condensing the information of input sentences according to the task and conveying the condensed information to both input and output sentences. We enabled information tokens to selectively attend to the tokens of input sentences through direct updates of the query vectors of the information tokens in each layer of the model. In addition, we enabled bidirectional operations for information tokens in the attention process in unidirectional language models, such as GPT(Radford et al., 2019), allowing information tokens to fulfill their roles even in unidirectional language models. Furthermore, we enhanced the performance by proposing an efficient structure that combines scaling and shifting within the layers to update the hidden state of the information tokens according to the task requirements.

Ultimately, when conducting experiments using IT-Tuning, we surpassed both full fine-tuning and LoRA(Hu et al., 2021) on the GLUE benchmark(Wang et al., 2018) using only 0.14M parameters, which is five times fewer. Furthermore, unlike existing methods, such as BitFit(Ben Zaken et al., 2022), PASTA(Yang et al., 2023), and $(IA)^3$(Liu et al., 2022a), where the increase or decrease in the number of parameters used for training is fixed and cannot be adjusted, making them applicable only to specific tasks (e.g., NLU tasks), IT-Tuning enables the adjustment of the number of parameters used for training through information tokens. Owing to the scalability of IT-Tuning, when applied to NLG tasks, we achieved a performance surpassing both full fine-tuning and Prefix Tuning(Li and Liang, 2021), as well as LoRA. This demonstrates the wide applicability of IT-Tuning for various tasks.

6 Limitation and Future work

One of the drawbacks of attention architecture is that the computational speed is significantly influenced by the length of the input. In our experiments, for NLU using BERT and RoBERTa, we utilized the existing [CLS] token as the Information Token without adding additional tokens, thus avoiding an increase in input length. However, for NLG tasks using GPT, additional tokens are added to serve as Information Tokens, resulting in an increase in input length. Given that our IT-Tuning has shown remarkably superior performance in NLG tasks, it remains the most efficient operation. However, while the number of parameters used in training is 10% less than LoRA, the training speed has increased by 10%, and the size of GPU memory used during training has increased by 100MB * batch size in the same setting as the experiments. As a future work to address these issues, we propose a method for NLG tasks where Information Tokens are selected from existing input tokens without adding additional tokens, similar to NLU tasks. Furturmore, we provide our implementation of IT-Tuning to support various future work : https://github.com/KU-INI/IT-Tuning.git

Acknowledgements

This work was supported by Institute for Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques).

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sashank Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny

Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2024. Palm: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24(1).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech Language*, 59:123–156.

Chin-Lun Fu, Zih-Ching Chen, Yun-Ru Lee, and Hung-yi Lee. 2022. AdapterBias: Parameter-efficient token-dependent representation shift for adapters in NLP tasks. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2608–2621, Seattle, United States. Association for Computational Linguistics.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. 2022. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123.

Baohao Liao, Yan Meng, and Christof Monz. 2023. Parameter-efficient fine-tuning without introducing new latency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4260, Toronto, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. AdaMix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5744–5760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xiaocong Yang, James Y. Huang, Wenxuan Zhou, and Muhao Chen. 2023. Parameter-efficient tuning with special token adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 865–872, Dubrovnik, Croatia. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Appendix

The Table 4 shows detailed hyperparameters used in our experiments.

Model	Dataset	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE
	Optimizer	AdamW							
	Warmup rate	0.06							
	LR Schedule	Linear							
RoB$_{large}$	Batch Size	8	64	8	8	32	32	64	16
	# Epochs	100	20	80	50	10	20	20	80
	Learning Rate	9e-4	7e-4	3e-4	9e-4	7e-4	5e-4	7e-4	5e-4
	IT-Tuning r	1							
	IT-Tuning a	1							
	Max Seq. Len.	128 + r							

Model	Dataset	E2E NLG Challenge
	Optimizer	AdamW
	Warmup rate	0.06
	LR Schedule	Cosine Restarts
GPT2$_{large}$	Batch Size	8
	# Epochs	20
	Learning Rate	5e-3
	IT-Tuning r	4
	IT-Tuning a	2
	Max Seq. Len.	128 + r
	Num Beam	10
	No Repeat Ngram	5
	Length Penalty	1.2

Table 4: The hyperparameters we used for RoBERTa and GPT2.