Unleashing the Potential of Compact Language Models: A Context-Optimized Soft Prompting Approach

Zhanxu Jiang

Institute of Artificial Intelligence, Beihang University, Beijing, China

jiangzhanxu@buaa.edu.cn

Abstract. The field of Natural Language Processing (NLP) has seen remarkable advancements with the development of large pre-trained language models, which excel in various tasks, especially through in-context learning. However, the increasing size of these models presents significant challenges for widespread deployment, particularly in resource-constrained environments. This study introduces Context-Optimized Soft Prompts (COSP), a new approach that can improve the performance of smaller language models in few-shots learning scenarios. COSP uses information from the presentation to initialize soft prompts, effectively addressing the limitations of smaller models when performing contextual learning. COSP is evaluated on multiple tasks in the SuperGLUE benchmark and showed significant performance improvements. Results show that COSP not only enhances model performance but also generates more diverse and evenly distributed soft prompts, contributing to robust and generalizable model behavior. Additionally, COSP accelerates the training process, potentially reducing computational resources for model adaptation. By enabling smaller models to perform complex tasks competitively, COSP opens up new possibilities for deploying complex language understanding techniques in resource-constrained environments.

Keywords: Pretrained Language Models, Prompt Tuning, In-context Learning.

1. Introduction

The evolution of natural language processing (NLP) has been significantly driven by large pre-trained language models (PLMs), which excel in a variety of tasks. Traditionally, fine-tuning these models for specific tasks has been the standard approach to achieving top performance. However, the extensive computational demands of fine-tuning, especially for large models, have led researchers to explore more efficient alternatives. Among these, prompt tuning has emerged as a promising method, enabling the use of PLMs with minimal task-specific adjustments [1-4].

Soft prompts have become a research hotspot because of their flexibility and efficiency. Soft prompts are continuous, learnable representations of vectors that guide the behavior of the model by optimizing those vectors without changing the parameters of the original model. Research in this area has explored various methods, starting with prefix-tuning [2], which involves adding trainable tokens to the input sequence, demonstrating efficient task adaptation with reduced fine-tuning requirements. Besides, P-Tuning [3] and its advanced version, P-Tuning v2 [4], add continuous prompts at different locations in the model, achieving results comparable to or even exceeding fine-tuning on natural language understanding (NLU) tasks. [5] extends this concept to few-shot learning, showing that prompt tuning can achieve competitive performance with minimal data. Additionally, researchers delve into the underlying mechanisms, exploring how prompts interact with the model's different components, contributing to the overall understanding of prompt tuning [6-8].

Large models, such as GPT-3, have demonstrated remarkable capabilities in various NLP tasks, particularly through In-context Learning (ICL), where they can adapt to new tasks without fine-tuning the parameters in the model. However, the exceptional performance of these models comes at a significant cost: this ability only emerges when the parameters come to a large number [9], thus requiring immense computational resources for inference, making them impractical for many pragmatic applications. This significant performance disparity between large and small models highlights the pressing need to enhance the capabilities of small language models to bridge this performance gap.

Inspired by in-context learning, this study introduces Context-Optimized Soft Prompts (COSP), a new approach designed to enhance the performance of smaller language models by leveraging information from demonstrations. By doing so, the approach addresses the limitations of small models in performing context learning, bridging the gap between the ICL capabilities of large language models and the practical constraints of deploying small models across a variety of NLP tasks in a way of few-shots learning. Through comprehensive experiments on the SuperGLUE benchmark dataset, it is demonstrated that COSP significantly improves the performance of smaller models, especially in few-shots learning scenarios. In addition, the analysis of experimental data shows that this method can obtain more soft prompts and accelerate the convergence process of training.

2. Methods

2.1. Intuition and Empirical Preliminaries

In-context learning has emerged as a powerful paradigm in natural language processing, demonstrating remarkable capabilities in task adaptation without the need for parameter updates. However, this ability is predominantly observed in large language models such as GPT-3 or even larger models [9]. The effectiveness of ICL appears to be closely tied to model scale, with smaller models often struggling to exhibit comparable in-context learning abilities. This scale dependency of ICL presents a significant challenge in deploying these capabilities in resource-limited environments that require more efficient models.

Soft prompts offer a promising avenue to address this challenge. Unlike traditional discrete prompts, soft prompts provide a mechanism to learn continuous representations that can capture latent semantic information from task-specific examples. These learned representations can potentially guide smaller models in a manner similar to how large models utilize in-context examples. In this way, smaller models are provided with a form of "compressed knowledge" that is tailored to the specific task at hand.

The methodology in this study is grounded in recent advancements in the understanding of soft prompts and ICL for PLMs. A novel perspective posits that prompts essentially stimulate the distributed capabilities among neurons within PLMs [10]. Another interesting research conceptualizes of ICL as a form of higher-level fine-tuning [11]. Building upon these insights, this study proposes a novel method that leverages information from demonstrations to initialize soft prompts. The approach aims to capture and distill task-specific information embedded in these demonstrations, transforming it into effective soft prompts. By doing so, this method addresses the limitation of smaller models in performing in-context learning (ICL), bridging the gap between the ICL capabilities of large language models and the practical constraints of deploying smaller models in various NLP tasks.

2.2. Context-Optimized Soft Prompts

The key innovation of this approach lies in its ability to harness the implicit knowledge contained in in-context examples to create more effective soft prompts. Formally, let $\mathcal{M}_{\psi}$ be a pre-trained language model with a vocabulary size of $|\mathcal{V}|$ and parameters ψ. Let $\{(\mathbf{x}_i, \mathbf{y}_i)\}_i$ be a labeled dataset for an NLU task, where $\mathbf{x}_{0:n} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is an input consisting of a sequence of discrete tokens, and $\mathbf{y}_i \in \mathbb{Y}$ is the corresponding label. The goal is to estimate the conditional probability for classification $f_{\mathcal{M}}(x) =$

$p(\hat{y}|x)$ with most of the parameters of $\mathcal{M}_{\psi}$ froze and the additional parameters finetuned. In the COSP framework, an encoder E_{θ} (LSTM or MLP) with parameters θ that processes a set of m few-shots learning examples $\{(\mathbf{x}_c, \mathbf{y}_c)\}_m$ is introduced to extract features in the demonstrations. The encoder outputs a feature representation: $\boldsymbol{h}_c = E(\{(\mathbf{x}_c, \mathbf{y}_c)\}_m)$.

Then these encoded features $\boldsymbol{h}_c$ are added as a prefix to the input embeddings. The COSP template is thus: $\boldsymbol{T}_{\text{COSP}} = \{\boldsymbol{h}_c\}$. The final input to the pre-trained language model is a sequence of embeddings: $Emb = \{\boldsymbol{T}_{\text{COSP}}, \boldsymbol{e}(\boldsymbol{x_0}), \boldsymbol{e}(\boldsymbol{x_1}), \ldots, \boldsymbol{e}(\boldsymbol{x_n})\}$, where $\mathbf{e} \in \mathbb{R}^{|\mathrm{V}| \times \mathrm{d}}$ is the embedding function of the pre-trained model.

During training, the parameters of the embeddings are optimized to minimize a task-specific loss function $\mathcal{L}$:

$$\min_{Emb} \ \mathcal{L}(f_{\mathcal{M}}(\boldsymbol{T}_{\text{COSP}}, \mathbf{x}), \mathbf{y}) \tag{1}$$

This design allows COSP to directly utilize features extracted from in-context learning examples as a continuous prompt prefix. By placing these features at the beginning of the input sequence, COSP potentially improves performance even for smaller models where traditional in-context learning might not be effective. The absence of discrete prompts simplifies the approach and allows the model to rely entirely on the learned continuous representations from in-context examples.

3. Results

3.1. Datasets and Metrics

To evaluate the effectiveness of COSP, experiments are conducted on the SuperGLUE benchmark [12]. SuperGLUE is a comprehensive suite of challenging NLU tasks consisting of eight diverse tasks that cover a wide range of linguistic phenomena and reasoning capabilities, from simple classification to complex reasoning tasks.

Specifically, three tasks: BoolQ, RTE, and WiC are selected in this study because previous methods [3] often don't perform as well on these subsets as other subsets. BoolQ [13] is a question answering task that tests a model's ability to understand yes-or-no questions in context. The Recognizing Textual Entailment (RTE) task challenges models to determine whether a given hypothesis can be inferred from a provided text. Word-in-Context (WiC) [14] evaluates a model's capacity to identify the meaning of polysemous words in different contexts.

To assess performance across these diverse tasks, accuracy is employed as the evaluation metric for all three tasks. Following the standard practice in related works, the average scores across all tasks as the overall scores are also listed.

3.2. Configuration

To evaluate the effectiveness of whether small models could learn patterns from the given context, the COSP method is mainly implemented on BERT model. Comparison methods include Fine-tuning and P-Tuning [1]. Since this method does not add additional parameters to each layer of the model, P-Tuning v2 is not selected for comparison. Regarding the ICL sample template format, this study follows the design of [15]. Specifically, the AdamW optimizer was used, and a batch size of {16, 32} and a learning rate of {2e-5, 1e-5, 5e-6}, accompanied by a learning rate decay technique, were used for training. A total of 5-20 epochs were trained, depending on the different sizes of datasets.

3.3. Results and Analysis

In this subsection, experimental data for the COSP method and the comparison method on several subsets of SuperGLUE are presented. At the same time, some soft prompts obtained by using COSP method are analyzed, trying to explain the reason why this method gets better performance.

3.3.1. Results. Table 1 presents the performances of COSP and comparing methods on the selected subsets of SuperGLUE benchmark. As can be seen from the table, COSP performs best on BoolQ and

RTE, while is closer to the finetuning results on WiC. It can be observed that the performance improvement of COSP on datasets with a large amount of data is not as good as fine-tuning, but the improvement on datasets with limited training data is greater. After calculating the average metrics, the COSP method also outperforms the other methods considered.

Table 1. Results on SuperGLUE benchmark. (**bold:** the best; underline: the second best)

	Method	BoolQ	RTE	WiC	Avg.
BERT-Base	Fine-Tuning	72.9	68.4	**71.1**	70.8
	P-Tuning	73.9	71.1	68.8	71.3
	COSP	**74.9**	**71.8**	70.3	**72.3**
BERT-Large	Fine-Tuning	77.7	74.8	**74.9**	75.8
	P-Tuning	77.2	74.4	72.7	74.8
	COSP	**78.4**	**75.3**	74.0	**75.9**

3.3.2. Visualization. Figure 1 shows the t-SNE visualizations of soft prompts, representing prompt distributions for the proposed method (left) and the random-initialized P-Tuning [6] (right), respectively. COSP shows a more decentralized clustering structure. These clusters vary in size and shape and are distributed throughout the two-dimensional space. In contrast, the random method also shows a certain clustering trend, but the boundaries between clusters are more separated. This feature implies that COSP can generate more diverse and differentiated prompts. This difference suggests that the proposed approach may better capture the continuous semantic meanings rather than extracting discrete prompts. The more concentrated distribution on the right may indicate that the P-Tuning method is not taking full advantage of continuity even when continuous representations are used. In contrast, COSP seems to make better use of the properties of continuous representation.

Moreover, the clusters in the left picture come in a variety of shapes and sizes, from tight ovals to loose irregular shapes. The cluster shape in the right picture is relatively uniform, with a flat oval or circle, which may mean that prompt expression ability is limited. The rich expressiveness of continuous prompts may indicate that the method proposed in this study can produce more diverse and structured distributions, exploring more possible representations in continuous space.

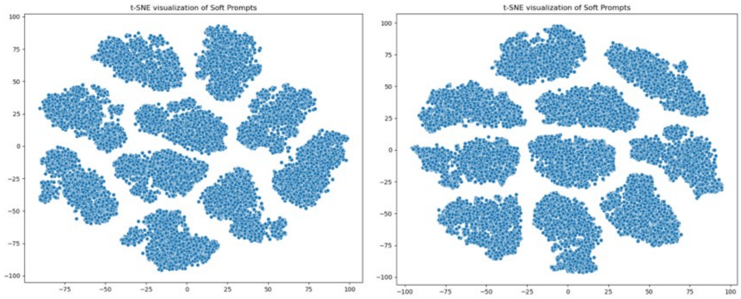

Figure 1. Visualization of soft prompts. The one on the left is COSP and the one on the right is random initialization.

3.3.3. Optimization process. Figure 2 shows the loss changes of two methods, random-initialized P-Tuning and COSP, in the early part of training process. The data from this training phase provided us with important insights into the performance characteristics of both methods. From the experimental results, COSP not only shows superior performance, but also provides us with a new perspective to understand the model optimization process.

First of all, it is obvious from the training curve that COSP shows a faster convergence rate than P-Tuning at the early stage of training. Whether it is training losses or validation losses, COSP is able to achieve lower levels in fewer training steps. This phenomenon not only shows that COSP can find a good parameter space faster, but also suggests that it may have higher parametric efficiency. More notably, COSP's validation losses are consistently lower than P-Tuning's, which is particularly important, showing that COSP is also better able to generalize to previously unseen data. Another noteworthy phenomenon is the smoothness of the COSP training curve. Compared with P-Tuning, the loss curve of COSP oscillates less and shows a more stable downward trend. From another perspective, COSP can be seen as a more efficient approach to soft prompts initialization. Previous methods usually rely on random initialization, class labels or sampled vocabulary. COSP, on the other hand, provides a task-specific, more informative starting point. This initial point brings the model closer to the optimal solution for the target task from the start, speeding up the convergence process and improving the final performance.

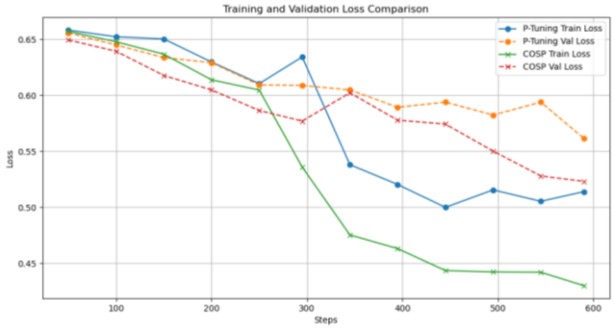

Figure 2. The loss figure of random-initialized P-tuning and COSP.

4. Discussion

While both COSP and P-Tuning use continuous prompts, COSP is more similar to prefix-tuning by setting the positions of prompts to prefixes, rather than involving different positions in the sequence as in p-tuning. This is also in line with the intuition of ICL to give the model a hint in advance. However, in terms of the choice of updatable parameters, COSP is different from prefix-tuning and P-Tuning v2 but more similar to P-Tuning, selecting the input embedding layer as the update part of the model, in order to catch up with the effect of fully fine-tuning with less training overhead.

The success of COSP has important implications for the popularization of large language modeling technologies, making advanced capabilities more accessible in resource-constrained environments. It provides an efficient way to adapt smaller models to new tasks without requiring extensive fine-tuning or the computational resources required for ICL in larger models. There are limitations, however. Further research is needed to explore optimizing presentation design and investigate the effectiveness of COSP in different model architectures. Future work could focus on applying COSP to multitasking learning, combining it with other cue engineering techniques. Going forward, if better performance is desired, trainable parameters can be extended to each layer of the model or to specific layers to increase the deep learning ability for specific tasks.

5. Conclusion

In this paper, a new method Context-Optimized Soft Prompts (COSP) that extract features from text demonstrations in the format of continuous prompts is presented. Overall, COSP not only improves the performance of small models, such as BERT, on natural language understanding tasks, but also accelerates and stabilizes the convergence process of prompt tuning. COSP effectively addresses the limitation of smaller models in performing in-context learning, particularly in few-shot learning scenarios. While there are areas for further investigation, including scalability and theoretical understanding of the method, COSP represents a significant step towards more accessible parameter-efficient fine-tuning solutions.

References

[1] Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.

[2] Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190.

[3] Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., & Tang, J. (2021). P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602.

[4] Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., & Tang, J. (2023). GPT understands, too. AI Open.

[5] Gu, Y., Han, X., Liu, Z., & Huang, M. (2021). Ppt: Pre-trained prompt tuning for few-shot learning. arXiv preprint arXiv:2109.04332.

[6] Oymak, S., Rawat, A. S., Soltanolkotabi, M., & Thrampoulidis, C. (2023, July). On the role of attention in prompt-tuning. In International Conference on Machine Learning (pp. 26724-26768). PMLR.

[7] Wei, C., Xie, S. M., & Ma, T. (2021). Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. Advances in Neural Information Processing Systems, 34, 16158-16170.

[8] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys, 55(9), 1-35.

[9] Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., ... & Sui, Z. (2022). A survey on in-context learning. arXiv preprint arXiv:2301.00234.

[10] Su, Y., Wang, X., Qin, Y., Chan, C. M., Lin, Y., Wang, H., ... & Zhou, J. (2021). On transferability of prompt tuning for natural language processing. arXiv preprint arXiv:2111.06719.

[11] Dai, D., Sun, Y., Dong, L., Hao, Y., Ma, S., Sui, Z., & Wei, F. (2022). Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. arXiv preprint arXiv:2212.10559.

[12] Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., ... & Bowman, S. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. Advances in neural information processing systems, 32.

[13] Clark, C., Lee, K., Chang, M. W., Kwiatkowski, T., Collins, M., & Toutanova, K. (2019). BoolQ: Exploring the surprising difficulty of natural yes/no questions. arXiv preprint arXiv:1905. 10044.

[14] Pilehvar, M. T., & Camacho-Collados, J. (2018). WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. arXiv preprint arXiv:1808.09121.

[15] Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., & Hajishirzi, H. (2022). Self-instruct: Aligning language models with self-generated instructions. arXiv preprint arXiv: 2212.10560.