XPrompt: Exploring the Extreme of Prompt Tuning

Fang Ma[1], Chen Zhang[1], Lei Ren[2], Jingang Wang[2]*, Qifan Wang[3], Wei Wu[2], Xiaojun Quan[4], Dawei Song[1]*

[1]Beijing Institute of Technology, [2]Meituan, [3]Meta AI, [4]Sun Yat-Sen University, *Corresponding author

Background

1.PLMs have achieved remarkable success in various NLP tasks with the pre-train-then-fine-tune paradigm.

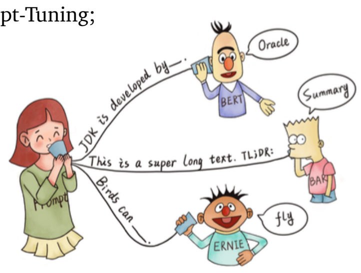
2.Fine-tune is parameter-inefficient for large scale PLMs:

The memory footprint is proportional to the number of trainable parameters.

3. With the development of GPT-3, prompt learning has drawn much attention in the NLP community:

From discrete prompt to continuous prompt, i.e., Prompt-Tuning;

Becoming more parameter efficient;



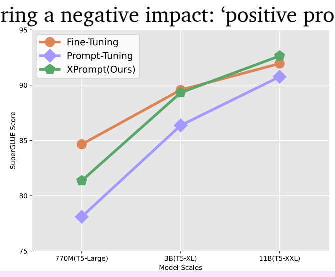
Motivation

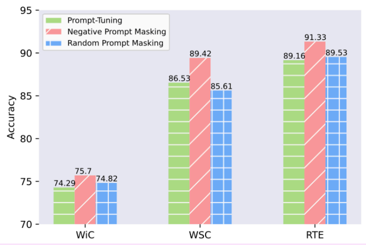
1.Prompt-tuning provides a parameter-efficient alternative to fine-tuning, which prepends a soft prompt to the input and only updating the parameters of prompt tokens during tuning.

2.Prompt-tuning achieves competitive performance to fine-tuning with the increase of the model scales.

3.However, there is still a large performance gap between prompt-tuning and fine-tuning for PLMs of smaller scales.

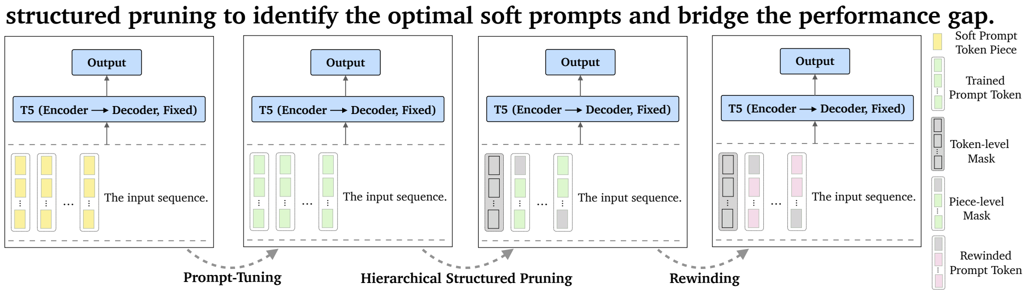
4.On a specific task, not all prompt tokens contribute equally to the task performance, while certain prompt tokens may bring a negative impact: 'positive prompt' and 'negative prompt' exist?





Method: XPrompt

Combining the idea of the lottery ticket hypothesis, we propose XPrompt with hierarchical structured pruning to identify the optimal soft prompts and bridge the performance gap.



XPrompt consists of three main stages:

(1)Prompt-Tuning:

Specifically, the prompt tuning learns an initial set of values for all soft prompt tokens on the target task.

(2)Hierarchical Structured Pruning:

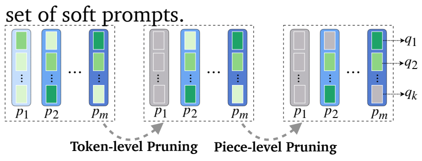
During the hierarchical structured pruning, token-level and piece-level pruning processes are repeatedly conducted to identify the optimal soft tokens and pieces at different compression ratios.

(3)Rewinding:

Finally, a weight rewinding technique is applied to re-train the soft prompts.

Hierarchical Structured Pruning

Hierarchical structured pruning is designed to separate negative prompt tokens from the trained prompt tokens, and identify an optimal set of soft prompts.



(1)Token-level Pruning

The token-level pruning is first used to identify negative prompt tokens. We associate mask variable γ_i to each soft prompt token vector p_i :

$$\hat{P}_e = \gamma \cdot P_e$$

where $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}, \gamma_i \in \{0,1\}$, and a 0 value indicates that the soft prompt token is pruned. We then calculate the importance score I_{p_i} of each token p_i to distinguish the negative prompt tokens from the others:

$$I_{p_i} = \mathbb{E}_{x \sim \mathscr{D}_x} \left| \frac{\partial \mathscr{L}(x)}{\partial \gamma_i} \right|$$

where $\mathscr{L}$ is the loss function and $\mathscr{D}_x$ is the training data distribution.

(2)Piece-level Pruning

However, the rest prompt tokens may still contain negative pieces. Thus, the piece-level pruning is then applied to identify more fine-grained negative prompt pieces within each prompt token. Mask variable ζ_i is associated with each piece in the soft prompt token to identify the negative prompt pieces:

$$\hat{q}_e = \zeta \cdot q_e$$

where $\zeta = \{\zeta_1, \zeta_2, \ldots, \zeta_k\}, \zeta_i \in \{0,1\}$, and 0 value indicates that the piece is pruned. We then calculate the importance score I_{q_i} of each piece for every prompt token embedding to prune the low-importance pieces.

$$I_{q_i} = \mathbb{E}_{x \sim \mathscr{D}_x} \left| \frac{\partial \mathscr{L}(x)}{\partial \zeta_i} \right|$$

We repeatedly conduct both token-level and piece-level pruning to obtain the sub-prompt tokens and pieces at different compression ratios.

Experiments

Datasets

We evaluate our method on various datasets of SuperGLUE benchmark in both high-resource and low-resource scenarios. Following previous works, we tune the prompt model on the training set for a fixed number of steps and report results on the validation set using the best checkpoint.

Baselines

Fine-Tuning: Standard fine-tuning approach of T5, all pre-trained parameters are fine-tuned on target task.
Prompt-Tuning: The vanilla prompt-tuning approach, which only tune the prepended soft prompt parameters.
P-Tuning: It is prompt-based method that uses the masked PLM to convert the target task into a close problem.
Prefix-Tuning: It prepends the prefix parameters to inputs of every transformer layer, only optimizes the prefix.

Results

Results on High-resource Scenarios

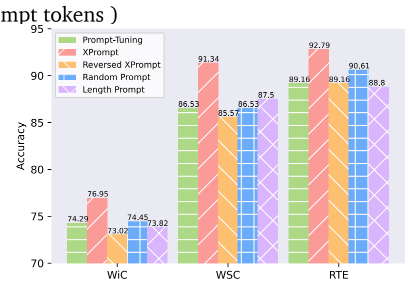
XPrompt significantly improves the performance of prompt tuning and helps close the gap with fine-tuning across all model scales. (Main experimental results (%) on seven SuperGLUE tasks.)

Model		WiC Acc	WSC Acc	CB Acc	COPA Acc	RTE Acc	Boolq Acc	MultiRC $F1_a$	Average Score
T5-Large 770M	Fine-Tuning*	73.50	88.50	94.30	72.0	90.60	88.30	85.40	84.65
	P-Tuning	70.37	64.42	92.85	76.0	79.78	83.02	79.96	78.06
	Prefix-Tuning	62.50	64.46	78.78	-	55.70	65.17	60.19	64.46
	Prompt-Tuning	72.25	68.26	82.14	76.0	85.19	83.02	79.86	78.10
	XPROMPT	**73.51**↑1.26	**70.39**↑2.13	**91.07**↑8.93	**82.0**↑6.0	**87.72**↑2.53	**83.82**↑0.8	**81.02**↑1.16	**81.36**↑3.26
T5-XL 3B	Fine-Tuning*	74.30	95.20	92.00	96.0	91.70	89.60	88.20	89.57
	P-Tuning	72.54	81.73	91.07	73.0	89.53	84.54	85.45	82.55
	Prompt-Tuning	74.29	86.53	91.07	91.0	89.16	87.58	84.89	86.36
	XPROMPT	**76.95**↑2.66	**91.34**↑4.84	**92.85**↑1.78	**95.0**↑4.0	**92.79**↑3.63	**89.00**↑1.42	**87.34**↑2.45	**89.32**↑2.96
T5-XXL 11B	Fine-Tuning*	78.50	95.20	100.00	99.0	92.10	90.40	88.60	91.97
	P-Tuning	76.80	94.23	92.85	93.0	89.80	86.98	87.56	88.75
	Prompt-Tuning	76.10	96.15	96.42	98.0	91.69	89.08	87.90	90.76
	XPROMPT	**77.69**↑1.59	**97.11**↑0.96	**100.00**↑3.58	**99.0**↑1.0	**94.94**↑3.25	**90.87**↑1.79	**88.90**↑1.0	**92.64**↑1.88

Results on Low-resource Scenarios

XPrompt performs much better in low resource scenarios. (The few-shot (32 samples) results (Acc, %) on three SuperGLUE tasks for the T5-XL model with 20 soft prompt tokens)

Model	Boolq	WiC	RTE
P-Tuning	64.99	54.23	57.40
GPT-3 XL1.3B‡	64.10	53.00	50.90
GPT-3 2.7B‡	**70.30**	51.60	56.30
PromptTuning	69.81	60.81	66.08
XPROMPT	70.23	**62.85**	**67.87**



Analysis and Discussion

Do Positive Prompts and Negative Prompts Exist?

We identify both positive and negative prompts through hierarchical structured pruning. For positive prompts, the first evidence is the large performance improvement of XPrompt over vanilla prompt tuning across all tasks and model scales. The second evidence is that the negative prompts perform worse than Prompt Tuning and XPrompt, as shown in the top right figure.

Parameter Efficiency

XPrompt is more parameter-efficient than Prompt-Tuning. (The number of tunable parameters comparison for T5-XL model with 20 prompt tokens.)

Model	WiC	WSC	CB	COPA	RTE	Boolq	MultiRC
Fine-Tuning	3×10^9	3×10^9	3×10^9	3×10^9	3×10^9	3×10^9	3×10^9
Prompt-Tuning	40960	40960	40960	40960	40960	40960	40960
XPROMPT	2560	6144	2560	15232	512	29184	27648
Percentage	6.25%	15%	6.25%	37.18%	1.25%	71.25%	67.5%

Granularity of Pruning and Prompt Length

Token-level pruning and fine-grained piece-level pruning are both important. And increasing prompt length (beyond 20) only yields marginal gains for XPrompt.

Model		WSC	CB	COPA	RTE
T5-Large	Prompt-Tuning	68.26	82.14	76.0	85.19
	Token-level	70.19	91.07	80.0	86.28
	Piece-level	69.23	89.28	79.0	86.64
	XPROMPT	**70.39**	**91.07**	**82.0**	**87.72**
T5-XL	Prompt-Tuning	86.53	91.07	91.0	89.16
	Token-level	89.42	92.85	93.0	92.41
	Piece-level	90.38	92.85	93.0	91.33
	XPROMPT	**91.34**	**92.85**	**95.0**	**92.79**

Length	Model	WSC	CB	COPA	RTE
10	Prompt-Tuning	82.69	87.50	87.0	88.44
	XPROMPT	87.50	91.07	93.0	90.61
20	Prompt-Tuning	86.53	91.07	91.0	89.16
	XPROMPT	91.34	92.85	95.0	92.79
100	Prompt-Tuning	89.42	91.07	90.0	89.16
	XPROMPT	91.94	92.85	94.0	92.79

Prompt Initialization and Transfer

Prompt initialization plays an important role in XPrompt. XPrompt Transfer can lead to performance gains.

Initialization Methods		WSC	COPA
Prompt-Tuning(SampledVocab)		86.53	91.0
XPrompt Initialization	RandomUniform	88.61	93.0
	SampledVocab	91.34	95.0

TransferMethod	WSC	⇔	COPA
TaskTransfer	86.53		92.0
XPromptTransfer_o	86.93		95.0
XPromptTransfer	91.40		98.0

Conclusions

Conclusions

1.This paper aims to close the large performance gap between prompt tuning and fine-tuning, especially for models of small and moderate scales.

2.By exploring the lottery ticket hypothesis in the context of prompt tuning, we have proposed a novel hierarchical structured pruning approach, namely XPrompt, to separate the positive prompts from the negative ones at both token-level and piece-level.

3.Extensive experimental results have demonstrated that XPrompt yields a more parameter-efficient prompt at an extremely small scale, yet with a competitive performance in effectiveness.

Acknowledgments

We thank the anonymous reviewers and chairs for their constructive suggestions. This research was supported in part by Natural Science Foundation of Beijing (grant number: 4222036) and Huawei Technologies (grant number: TC20201228005).

The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)

dwsong@bit.edu.cn