Selective Prefix Tuning for Pre-trained Language Models

Hongyi Zhang[1†], **Zuchao Li**[1*†], **Ping Wang**[2,3], **and Hai Zhao**[4]
[1]National Engineering Research Center for Multimedia Software,
School of Computer Science, Wuhan University
[2]School of Information Management, Wuhan University
[3]Key Laboratory of Archival Intelligent Development and Service, NAAC
[4]Shanghai Jiao Tong University
{harryzhang,zcli-charlie,wangping}@whu.edu.cn, zhaohi@cs.sjtu.edu.cn

Abstract

The prevalent approach for optimizing pre-trained language models in downstream tasks is fine-tuning. However, it is both time-consuming and memory-inefficient. In response, a more efficient method called Prefix Tuning, which inserts learnable vectors into each Transformer layers, has been proposed and proven effective. Recent investigations reveal that prefix tokens carry context-specific information, prompting the hypothesis that enhancing their specialization can improve model performance. To address this, we propose Selective Prefix Tuning (SPT), integrating a selective mechanism inspired by selective self-attention. Additionally, we introduce Selective Loss (SL) to encourage diversity in prefix tokens. Extensive experiments validate the effectiveness of SPT in sentence and token classification tasks. We contribute insight into understanding the role of prefix in model adaptation.

1 Introduction

Fine-tuning serves as a pivotal mechanism to adapt large pre-trained models for downstream tasks by adjusting all parameters but it is prohibitively expensive. Parameter-efficient learning is an emerging framework that freezes pre-trained models and only tunes a few number of task-specific parameters. An exemplary illustration of this paradigm is Prefix Tuning and P-tuning-v2 (Li and Liang, 2021; Liu et al., 2022a), wherein a fixed-length of learnable vectors is concatenated in the Transformer layer. Studies have proven that Prefix Tuning can achieve comparable performance or even outperforms fine-tuning (Liu et al., 2022b).

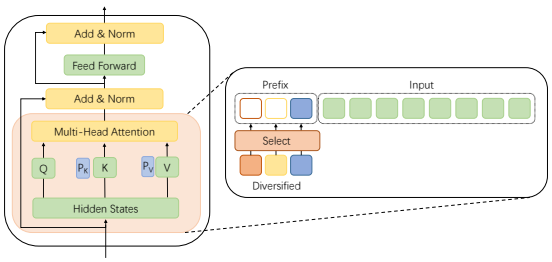

Figure 1: An illustration of SPT where the left is Transformer architecture with prefix and the right is our proposed SPT.

Recent investigations into soft tokens have unveiled that they harbor domain- or context-specific words rather than a general instruction for certain tasks (Lester et al., 2021). For instance, when tuning model on BoolQ (Clark et al., 2019) benchmark, where approximately 20% of the questions in the training dataset pertain to the "Nature/Science" field, the learned prefix exhibits a notable frequency of vectors whose nearest neighbors are words like "science". Ju et al. (2023) assumes that a soft token is a combination of discrete tokens and surprisingly finds that a lot of tokens in the prefix is task-irrelevant, but serves to induce the model for the correct output.

Although these studies are based on tuning only the prefix tokens in the input layer, it is reasonable to generalize this to methods that insert prefix tokens into multiple Transformer layers. It prompts the reasonable assumption that the role of prefix is to furnish the model with specific contextual cues. Intuitively, making the prefix tokens more "specialized" can help model focus on the useful prefix cues and ignore unrelated ones. Besides, the prefix should be diverse so as to provide the model with a richer bank of context. However, previous studies didn't fully exploit these attributes. So in this work, we posit that enhancing the specializa-

*Corresponding author.
†Equal contribution.

This work was supported by the National Natural Science Foundation of China (No. 62306216, No. 72074171, No. 72374161), the Natural Science Foundation of Hubei Province of China (No. 2023AFB816), the Fundamental Research Funds for the Central Universities (No. 2042023kf0133).

Findings of the Association for Computational Linguistics: ACL 2024, pages 2806–2813

tion of prefix tokens and forcing them to capture diversified information can lead to improved model performance.

To fulfill the first objective of refining the specificity of the prefix, we guide the model to select a subset of useful prefix tokens. Inspired by Geng et al. (2020), which employs an auxiliary network to mask out less relevant tokens, we introduce a selective mechanism into Prefix Tuning, presenting our novel method called Selective Prefix Tuning (SPT). We strive to avoid adding new parameters, so we simplify the method by leveraging the original attention scores to effectively mask out irrelevant tokens in the prefix.

The second objective is dedicated to augmenting the diversity of the prefix through the incorporation of Selective Loss (SL) as a regularization term. Drawing inspiration from Li et al. (2018), where explicit terms are introduced to foster diversity among attention heads, our approach calculates the absolute value of average cosine similarities between prefix tokens to formulate the Selective Loss.

Through extensive experiments focusing on sentence and token classification tasks under full data settings, we validate our approach. Besides, by adjusting newly introduced hyper-parameters, we further study the effect of the Selective Mask and find the Selective Loss can help the model learn more features and sometimes even achieve lower training loss. Our code is available at https://github.com/potter-Zhang/Selective-Prefix-Tuning.

2 Related Works

Parameter Efficient Fine Tuning Since pre-trained models are getting larger, fine-tuning is prohibitively expensive. In response to this, the framework of Parameter-efficient learning, which freezes the pre-trained model and tunes a small number of parameters, is proposed. For instance, P-tuning (Liu et al., 2022a) and Prefix Tuning (Li and Liang, 2021) are methods that insert soft prompts in inputs or hidden states. P-Tuning v2 (Liu et al., 2022b) extends Prefix Tuning to Natural Language Understanding tasks. There are also methods based on the gradient like SIFT (Song et al., 2024), which uses sparse gradients to reduce memory consumption.

Prompt-tuning methods Prompt-tuning-related methods have achieved notable success. Some researchers even use soft prompts for multi-modal learning (Yang et al., 2024). In order to incorporate soft tokens, various approaches have been explored. For instance, certain methods incorporate trainable tokens directly into the input layer (Lester et al., 2021). Others add trainable bias to the hidden states of corresponding prefix tokens (Li et al., 2023), and some opt to train the entire key-value (KV) cache of the Transformer model (Liu et al., 2022b). The baseline method employed in this paper is P-Tuning-v2 (Liu et al., 2022b), which focuses on training the KV cache.

Selective self-attention network With an overlap of our motivation, SSAN(Geng et al., 2020) uses a selector module to select a subset of tokens and apply self-attention afterwards, so the model can put more attention weight on the content words. Hu et al. (2022) introduces adaptive threshold into selective self-attention network when solving Chinese NER problems. However, our method applies selective mechanism only in learnable prefix and it is used in every layer while SSAN only masks out tokens in the first layer and introduces extra parameters for selecting.

Disagreement regularization While the use of multiple attention heads enables a distinct focus on various segments of the sequence, there is no guarantee that different attention heads will learn different features. Li et al. (2018) introduces three types of disagreement terms to explicitly encourage the model to attend to different features. Inspired by this, our approach with Selective Loss is designed to steer the prefix towards acquiring intricate and diverse contextual information.

3 Methodology

3.1 Prefix Tuning

We first briefly recap the structure of Transformer (Vaswani et al., 2023). A Transformer layer is a block consisting of multi-head attention and a fully connected feed-forward network. Formally speaking, a Transformer block is calculated as follows:

$$Attn(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \quad (1)$$

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2 \quad (2)$$

Prefix Tuning (Li and Liang, 2021) prepends trainable tokens to each Transformer layer. Let $P_k, P_v \in \mathbb{R}^{l_p \times d}$ be the keys and values prefix respectively, where l_p denotes the length of prefix

and d is the dimension, the self-attention block can be calculated as below:

$$Attn(Q, K', V') = softmax(\frac{QK'^T}{\sqrt{d}})V' \quad (3)$$

Where $K' = [P_k; K]$ and $V' = [P_v; V]$. [;] denotes concatenation function.

3.2 Selective Prefix Tuning

Previous studies have shown that prefix can provide certain context cues to the models but they haven't exploited this ability. So we introduce two novel methods to improve the performance of Prefix Tuning. In section 3.2.1 we introduce Selective Mask to further ignore unrelated prefix. In section 3.2.2, we show how we construct the Selective Loss to improve diversity of prefix tokens so as to better capture different context information.

3.2.1 Selective Mask

The core idea here is to generate a mask to ignore prefix tokens unrelated to the current content. By masking irrelevant tokens the model can pay more attention to content words that contribute to the meaning of the sentence and thereby improving its performance.

Consider a token t in sequence, since the core mechanism of self-attention layer consists of multiple heads working in parallel, the head will calculate similarities with prefix, i.e., any token r in the prefix:

$$s_{tr} = \frac{x_t^T W_q^T (P_k)_r}{\sqrt{d}} \quad (4)$$

where W_q is query matrix, $(P_k)_r$ is token r in keys prefix and x_t is the tth token. To ignore unimportant information in prefix, we apply a soft mask for each head in the prefix. A soft masking function is a non-decreasing function that maps attention scores to a value in (0, 1). We take the following masking function $m(x)$:

$$m(x) = sigmoid(\alpha x) \quad (5)$$

where α is an amplifier factor greater than one controlling the softness of the mask. The computed mask is then applied to the attention weights of the prefix:

$$a_{tr} = \frac{m(s_{tr})exp(s_{tr})}{\sum_{q=1}^{l_p} m(s_{tq})exp(s_{tq}) + \sum_{q=l_p+1}^{l_s} exp(s_{tq})} \quad (6)$$

here, l_s is the sequence length including the prefix. The first sum of denominator is the attention weights querying prefix, and the second one represents normal attention weights of tokens. The mask function is only applied to the prefix tokens. It should be noticed that here we apply the mask head-wise.

3.2.2 Selective Loss

Although Selective Mask can ignore unrelated prefix tokens, it does not guarantee prefix can learn diverse features. Inspired by disagreement regularization of attention heads(Li et al., 2018), which uses extra regularization terms to force the model to learn different features. We propose our Selective Loss to improve the expressive ability of prefix. This term is designed to force the prefix tokens to be orthogonal.

Specifically, for the prefix of keys, we first calculate the cosine similarity between vector pair $(P_k)_i^q$ and $(P_k)_j^q$, where $(P_k)_i^q$ represents the ith token in keys prefix of qth layer, using dot product of normalized vectors. In order to make vectors orthogonal, we then calculate the average absolute value of cosine similarity. Formally speaking, the regularization term in one Transformer layer can be expressed as below:

$$KSL^q = \frac{2}{l_p(l_p-1)} \sum_{i=1}^{l_p} \sum_{j=i+1}^{l_p} \frac{|(P_k)_i^q \cdot (P_k)_j^q|}{||(P_k)_i^q|| ||(P_k)_j^q||} \quad (7)$$

here l_p is the length of prefix. Similarly, the regularization term for values prefix can be defined as below:

$$VSL^q = \frac{2}{l_p(l_p-1)} \sum_{i=1}^{l_p} \sum_{j=i+1}^{l_p} \frac{|(P_v)_i^q \cdot (P_v)_j^q|}{||(P_v)_i^q|| ||(P_v)_j^q||} \quad (8)$$

where $(P_v)_i^q$ represents the ith tokens in values prefix of qth layer. Combining these two regularization terms and consider all Transformer layers, we get the Selective Loss:

$$SL = \frac{1}{2} \sum_{q=1}^{L} (KSL^q + VSL^q) \quad (9)$$

Where L is the number of Transformer layers. Finally, we combine Selective Loss with the original training objective and get our revised version:

$$L = L_{ori} + \lambda * SL \quad (10)$$

Model		SuperGLUE					
		BoolQ	COPA	RTE	WiC	WSC	Avg
BERT-base (110M)	FT	**72.9**	67.0	68.4	**71.1**	63.5	68.6
	PT-2	72.5	67.4	71.3	69.5	**65.4**	69.2
	SPT	72.5	**70.0**	**74.3**	70.7	**65.4**	**70.6**
BERT-large (335M)	FT	**77.7**	69.0	70.4	74.9	68.3	72.1
	PT-2	75.8	73.0	78.3	75.1	68.3	74.1
	SPT	75.8	**80.0**	**78.7**	**75.7**	**69.2**	**75.9**
RoBERTa-large (355M)	FT	**86.9**	**94.0**	86.6	**75.6**	**63.5**	**81.3**
	PT-2	84.8	93.0	89.5	73.4	**63.5**	80.8
	SPT	84.4	93.0	**89.9**	73.8	**63.5**	80.9

Table 1: The results on SuperGLUE development set. The metric is accuracy. Results for FT and PT-2 on BERT-large and RoBERTa-large are taken from Liu et al. (2022b). Results for FT on BERT-base are from Liu et al. (2022a), and results for PT-2 on BERT-base are from Zhang et al. (2023).(FT: vanilla fine-tuning; PT-2: P-Tuning v2; SPT: Selective Prefix Tuning; **bold**: the best score; underline: the second best)

Model	NER					
	CoNLL03			CoNLL04		
	FT	PT-2	SPT	FT	PT-2	SPT
BERT-base	-	89.3	89.8	-	82.6	82.8
BERT-large	92.8	90.2	90.0	85.6	84.5	85.3

Table 2: The results for NER datasets including CoNLL03 and CoNLL04. Results for FT and PT-2 on BERT-large are from Liu et al. (2022a). Results for PT-2 on BERT-base are from Zhang et al. (2023). The metric here is f1 score.

where λ is a hyper-parameter to control the ratio of Selective Loss. L_{ori} refers to the original loss in different tasks settings.

4 Experiments

4.1 Experimental Setup

For Natural Language Understanding, we conduct experiments on 5 tasks from SuperGLUE (Wang et al., 2020) benchmark including BoolQ (Clark et al., 2019), COPA (Gordon et al., 2012), RTE (Wang et al., 2018), WiC (Pilehvar and Camacho-Collados, 2019), WSC (Levesque et al., 2012) using three models including BERT-base / large (Devlin et al., 2019) and RoBERTa-large (Liu et al., 2019) instantiated by HuggingFace Transformers (Wolf et al., 2020). For Named Entity Recognition(NER) tasks, we conduct experiments on CoNLL03 (Tjong Kim Sang and De Meulder, 2003) and CoNLL04 (Carreras and Màrquez, 2004) with BERT-base / large (Devlin et al., 2019).

4.2 Results

We report the main results in Table 1. We observe that SPT achieves 1.4% improvement over P-Tuning v2 on SuperGLUE for BERT-base. For BERT-large, SPT surpasses P-Tuning v2 on SuperGLUE by 1.8%. For RoBERTa-large, SPT outperforms P-Tuning v2 on SuperGLUE by 0.1%. We also report results on NER datasets in Table 2. It can be seen that on CoNLL03 and CoNLL04, SPT consistently outperforms P-Tuning v2 and is even comparable to fine-tuning. Although on CoNLL03 we oberve a slightly performance drop, SPT achieves an average of at least 0.3 improvement with BERT-base and BERT-large models, indicating its superiority.

4.3 Ablation Studies

To further study the contribution of Selective Mask and Selective Loss to the improvement of model performance, We conduct experiments on 5 tasks from SuperGLUE with BERT-base model. The results are reported in Table 3.

Although on BoolQ dataset, Selective Mask and Selective Loss alone leads to slight decrease in performance, in most cases it can be seen that with Selective Loss or Selective Mask individually, the performance is improved. When combining the two components, greater improvement is observed.

Model	BoolQ	COPA	RTE	WiC	WSC
PT-2	72.5	67.4	71.3	69.5	63.5
PT-2 + SM	72.1	68.0	73.2	70.8	65.4
PT-2 + SL	72.1	69.0	74.0	71.0	65.4
SPT	72.5	70.0	74.3	70.7	65.4

Table 3: Results for ablation studies. The model we use is BERT-base. The metric here is accuracy. Results for PT-2 are from Zhang et al. (2023).

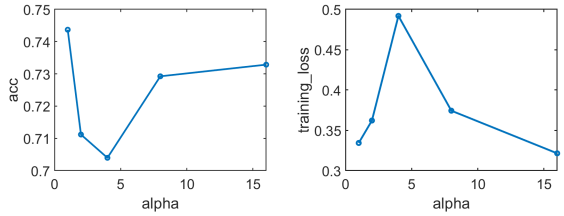

Figure 2: Accuracy on development set(left) and training loss(right) of RTE using BERT-base model with different values of α.

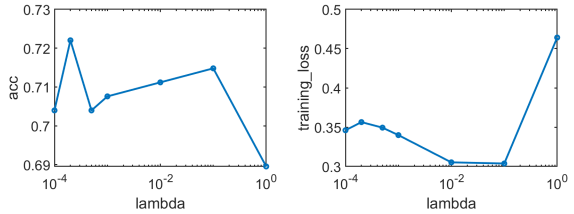

Figure 3: Accuracy on development set(left) and training loss(right) of RTE using BERT-base model using different values of λ.

4.4 Intrinsic Evaluation

We study the influence of different hyper-parameters in SPT. 4.4.1 studies the impact of amplifier factor α. 4.4.2 further probes into the influence of Selective Loss. We conducted experiments with BERT-base model on RTE dataset. We run totally 50 epochs with a learning rate of 1e-2 and a prefix length of 8. We modify α and λ in the following experiments to see the influence of these key hyper-parameters.

4.4.1 Selective Mask

We first investigate the influence of Selective Mask. With batch size of 8 and $\lambda = 2e-4$, we train our model with $\alpha = [1, 2, 4, 8, 16]$ and report the results in figure 2. It is observed that combined with SL, a proper α can improve the performance a lot. When α is smaller, it will be easier for the model to train since an α that is too large will make the function $sigmoid(\alpha x)$ become steeper, with a especially large gradient around the zero and almost no gradient for other numbers, which might lead to difficulties in optimization. But larger α can help the model focus more on useful prefix tokens by ignoring unrelated ones, bringing low training loss and a good generalization ability. In practice, a careful search of α is needed.

4.4.2 Selective Loss

With batch size of 16 and $\alpha = 8$, the best accuracy on development set and training loss on RTE with different values of lambda is reported in figure 3. It is observed that as λ increases, the accuracy shows the trend of increasing, but if λ is too large, a performance drop is observed, which is consistent with our knowledge of regularization. Besides, we notice that with the increase of lambda, training loss is actually going down, which validates that Selective Loss can help model learn more features. However, large values of λ will lead to underfitting of the model.

5 Conclusions

In this work, we propose a novel way to improve the performance of Prefix Tuning. Without adding extra parameters, significant improvements are observed. Experiments on Natural Language Understanding (NLU) and Named Entity Recognition (NER) tasks validate the effectiveness of our approach. In addition, Selective Prefix Tuning provide us with a new perspective of the role and attributes of pseudo prefix. It can be served as a promising method for Parameter-efficient learning.

Limitations

The proposed SPT is applicable to Large Language Models just like Prefix Tuning. However, due to limitation of computation resources, we only applied SPT to encoder-only models and didn't conduct extensive experiments with LLM or encoder-decoder models.

References

Xavier Carreras and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 89–97, Boston, Massachusetts, USA. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Xinwei Geng, Longyue Wang, Xing Wang, Bing Qin, Ting Liu, and Zhaopeng Tu. 2020. How does selective mechanism improve self-attention networks?

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Biao Hu, Zhen Huang, Minghao Hu, Ziwen Zhang, and Yong Dou. 2022. Adaptive threshold selective self-attention for Chinese NER. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1823–1833, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Tianjie Ju, Yubin Zheng, Hanyi Wang, Haodong Zhao, and Gongshen Liu. 2023. Is continuous prompt a combination of discrete prompts? towards a novel view for interpreting continuous prompts. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7804–7819, Toronto, Canada. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-head attention with disagreement regularization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2897–2903, Brussels, Belgium. Association for Computational Linguistics.

Jonathan Li, Will Aitken, Rohan Bhambhoria, and Xiaodan Zhu. 2023. Prefix propagation: Parameter-efficient tuning for long sequences. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1408–1419, Toronto, Canada. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022a. 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. 2022b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks.

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.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Weixi Song, Zuchao Li, Lefei Zhang, Hai Zhao, and Bo Du. 2024. Sparse is enough in fine-tuning pre-trained large language models.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Juncheng Yang, Zuchao Li, Shuai Xie, Wei Yu, Shijun Li, and Bo Du. 2024. Soft-prompting with graph-of-thought for multi-modal representation learning.

Zhen-Ru Zhang, Chuanqi Tan, Haiyang Xu, Chengyu Wang, Jun Huang, and Songfang Huang. 2023. Towards adaptive prefix tuning for parameter-efficient language model fine-tuning.

A Prefix Length

We also conduct experiments to investigate the influence of prefix length. Here α is 8 and λ is 2e-4. When prefix length increases, the final training loss decreases as shown in figure 4. Performance on validation set first increases as the prefix becomes longer, but after it reaches a threshold, longer prefix will lead to performance drop. Since the longer the prefix, the more parameters can be tuned, this indicates that overfitting occurs, which is consistent with the finding in Li and Liang (2021).

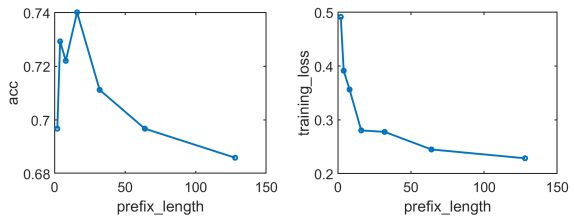

Figure 4: Accuracy(left) on development set and training loss(right) of RTE using BERT-base model with different values of prefix length. When prefix length increases, the model shows performance gain. After it reaches a threshold, increasing prefix length will lead to drop of performance.

B Visualization of Selective Mask

We visualize the change after applying the Selective Mask on the attention scores. From figure 5 we can see that the mask will almost ignore the tokens whose similarity is a negative value. The greater the alpha is, the tighten the bound of ignorance. As for the positive part, it is almost the same as the original one when attention scores is sufficiently large.

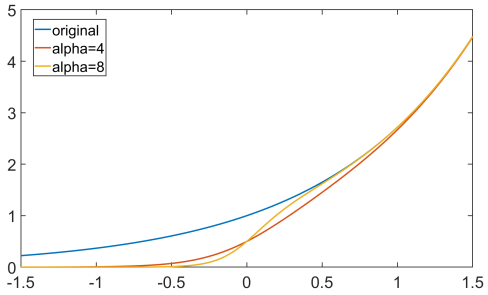

Figure 5: A visualization of the attention weight curve with Selective Mask. The x-axis represents the attention scores. The blue curve is the original calculation of attention weight, i.e. the exp(x) function. The rest are the calculation methods with Selective Mask of different values of α.