Instance-Guided Prompt Learning for Few-Shot Text Matching

Jia Du [1], **Xuanyu Zhang** [2], **Siyi Wang** [1], **Kai Wang** [1], **Yanquan Zhou** [1]*,
Lei Li [1]*, **Dongliang Xu** [2], **Qing Yang** [2]
[1]Beijing University of Posts and Telecommunications, Beijing, China;
[2]Du Xiaoman Financial, Beijing, China
{jia_du,zhouyanquan,leili,wk}@bupt.edu.cn
{zhangxuanyu,yangqing,xudongliang}@duxiaoman.com

Abstract

Few-shot text matching is a more practical technique in natural language processing (NLP) to determine whether two texts are semantically identical. Recent studies based on prompt learning have shown remarkable results. These methods primarily employ uniform prompt patterns for all instances. But they fail to take into account the connection between prompts and instances. This paper argues that dynamically strengthening the correlation between particular instances and the prompts is necessary because fixed prompts cannot adequately fit all diverse instances in inference. Thus, we propose IGATE: Instance-Guided prompt leArning for few-shoT tExt matching, a novel pluggable prompt learning method. The gate mechanism between embedding and encoder of the PLM makes use of the semantics of instances to regulate the effects of the gate on the prompt tokens. The experimental findings show that IGATE achieves SOTA performance on MRPC and QQP, outperforming strong baselines. Our codes are available at https://github.com/Du-Jia/IGATE.

1 Introduction

Prompt learning has made significant progress in few-shot text matching due to the widespread use of pre-trained language models (PLMs) in natural language processing (NLP). This approach reformulates the text matching task as a cloze-style question which requires PLMs to predict what the blank should be filled with. More specifically, after being added prompt tokens and masked tokens, a PLM is used to predict the masked words and map these words to the real labels. Recent studies have shown that prompt learning achieves better performance on few-shot text matching. For example, with only thirty-two examples in MRPC (Wang et al., 2019a), prompt learning achieves about 85% of the performance of fully supervised fine-tuning models (Zhou et al., 2022).

Brown et al. introduce the concept of prompt in the in-context method for the first time. Subsequently, Schick and Schütze propose PET which gains improvement by exploiting patterns in natural language understanding. Some studies (Gao et al., 2021; Shin et al., 2020; Zhong et al., 2021) search prompts automatically to reduce the reliance on human experts for manual pattern design. All of these methods use natural language as prompts thus they are named discrete prompts. Other methods such as P-tuning (Liu et al., 2021b), Prefix-tuning (Li and Liang, 2021), and P-tuning-V2 (Liu et al., 2021a) replace natural language prompts with trainable continuous tokens on this basis to automatically search for optimal prompts in a high-dimensional space. Correspondingly, these methods are also called continuous prompts.

However, current prompt learning methods typically train models for specific task goals with little regard to the applicability of samples to the prompt. Even though some recent works (Liu et al., 2022; Gu et al., 2021) attempt to use contextual information for generating prompts, they frequently ignore how samples affect prompts and concentrate on how prompts contribute to instances. In addition, they usually fix the prompts during inference so that all test samples share the same pattern.

To solve these problems, this paper proposes a novel method, IGATE: Instance-Guided prompt leArning for few-shoT tExt matching, to guide the construction of a prompt with instance semantics. We introduce the gate mechanism based on continuous prompts to regulate the flow of features in the prompts. Meanwhile, since the semantics of instances are used to construct the prompt, each instance can play a restrictive role in building the pattern. So gate mechanism alleviates the problem of limited adaptation of the same pattern to

* Co-corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3880–3886

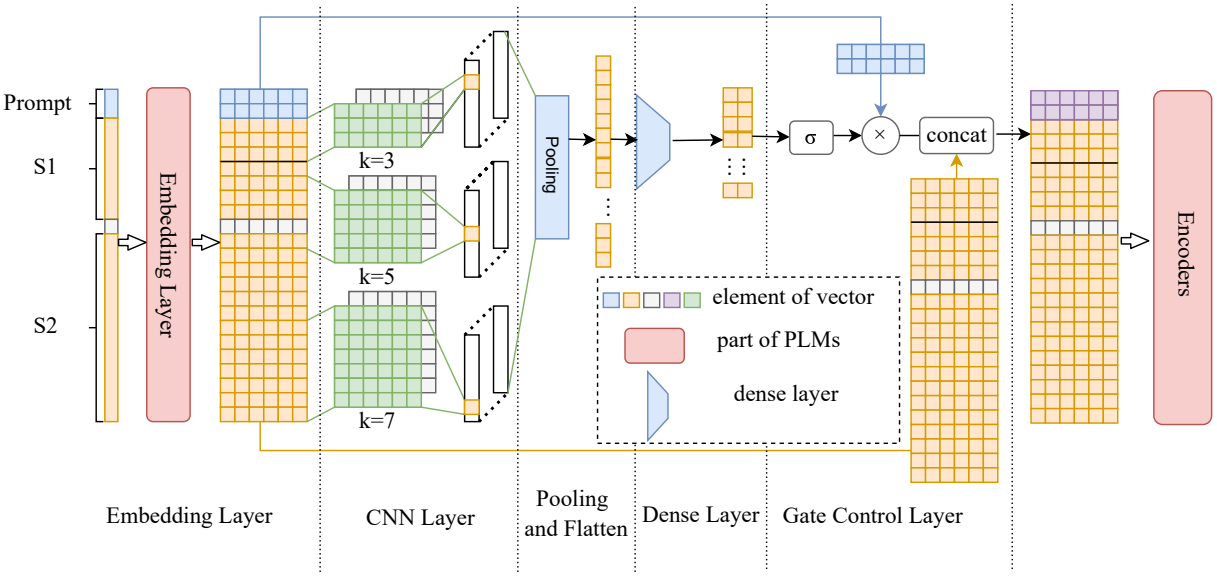

Figure 1: The architecture of IGATE.

instances in the reasoning process. In short, this study has the following contributions:

- We propose a novel approach to dynamically generate prompts for each instance by introducing gate mechanism that allows instances to participate in building prompts.
- Our method improves the performance of few-shot text matching ability of models by merely processing the embedding of PLMs. This method can be easily transferred to other prompt methods.
- Extensive experiments validate the effectiveness of IGATE on few-shot text matching. We improve the performance by 2.05 on average on MRPC and QQP. Meanwhile, we also verify that IGATE can be generalized to the natural language inference (NLI) task through experiments on corresponding datasets such as RTE, SNLI, and QNLI, which improve the results by 1.52 on average.

2 Methodology

In this section, we introduce conventional prompt learning approaches for few-shot text matching. Then we introduce the architecture of IGATE, which adds an additional layer between the embedding layer and the encoder of PLMs. The architecture is shown in Figure 1.

2.1 Prompt learning for few-shot text matching

Let $\mathcal{M}$ be a PLM with vocabulary $\mathcal{V}$. For instances $(\boldsymbol{s_1}, \boldsymbol{s_2})$ in few-shot text matching, our goal is to predict whether the text pair $(\boldsymbol{s_1}, \boldsymbol{s_2})$ is semantically identical, where $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$ are two sequences of tokens. In prompt learning, $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$ are usually put into a specific pattern, consisting of special tokens, text pairs, and external prompt tokens. For example, the method plugs the instance $(\boldsymbol{s_1}, \boldsymbol{s_2})$ into a pattern which contains prompt tokens $\mathcal{V}_p$: `[CLS]`, p_1, $\boldsymbol{s_1}$, `[MASK]`, p_2, $\boldsymbol{s_2}$, `[SEP]` and then uses $\mathcal{M}$ to select the appropriate word $w \in \mathcal{V}^*$, where $p_1, p_2 \in \mathcal{V}_p$ are prompt tokens, $\mathcal{V}^*$ is the set of candidate label words. Finally, label word w is mapped to the real label. Taking MRPC task as an example, the mapping function is usually "yes"$\rightarrow$ 1, "no"$\rightarrow$ 0:

$$\mathcal{P}(y|(\boldsymbol{s_1}, \boldsymbol{s_2})) = P(w|\mathcal{M}((\boldsymbol{s_1}, \boldsymbol{s_2}, \boldsymbol{p}))) \quad (1)$$

where $\mathcal{P}$ demonstrates the probabilities of y when given input text pair $(\boldsymbol{s_1}, \boldsymbol{s_2})$, $\boldsymbol{p} = p_1, p_2, ..., p_\ell$, ℓ is the length of prompt. Generally, prompt learning is dived into two groups: discrete and continuous. Discrete prompt learning methods search human-understandable prompt tokens, which means the prompt tokens $\mathcal{V}_p$ is a subset of the vocabulary of the PLM. Differently, continuous prompt learning methods take some pseudo-tokens in patterns, which are projected to differentiable high-dimensional vectors in the training or inference process.

$pattern_1$ id	$prompt_2$ id	# of different predictions	rate(%)
1	2	58/408	14.22
1	3	81/408	19.85
1	4	72/408	17.65
2	3	93/408	22.79
2	4	106/408	25.98
3	4	65/408	15.93

Table 1: The different number of instances in MRPC prediction results between two patterns. Here different prompt ids in the table represent different patterns, and the specific correspondence is as follows: "(id=1) $\boldsymbol{s_1}$ [mask] In fact $\boldsymbol{s_2}$ "; "(id=2) $\boldsymbol{s_1}$ [mask] This is the first time $\boldsymbol{s_2}$ "; "(id=3) $\boldsymbol{s_1}$, [mask] . $\boldsymbol{s_2}$ "; "(id=4) $\boldsymbol{s_1}$. [mask] However $\boldsymbol{s_2}$ ".

2.2 IGATE: Instance-guided propmt learning method

A single pattern cannot be adapted to all instances Previous prompt learning approaches ignore the fact that a pattern cannot fit all instances. Meanwhile, these works focus on searching for optimal prompts (discrete or continuous) for specific tasks. All instances have used the same prompts. These searched prompts have been fixed during the inference process. Thus, these approaches have been concerned with specific tasks but ignored the fact that there are huge variability between instances. As shown in Table 1, for the same task, the prediction results of four different patterns are very distinct, and the different data even account for up to 25% of the total data. This confirms our point: it is difficult for a single pattern to be valid for all instances.

Instance-guided prompt learning Previous continuous prompt learning methods usually optimize prompt tokens for text matching objectives. However, they ignore the influence of the instance on prompts, which means the prompt is the task-level prompt. Therefore, in order to make prompts better adaptable to different instances, we try to extend prompts to the instance-level in IGATE. Specifically, based on the task-level prompts, we expect the model to automatically select features in the prompt token which are applicable to the current instance and optimize that capability by the gradient. IGATE controls the flow of prompt information through a gate mechanism that depends on instance information, to assist the model to consider the semantic information of instances when constructing prompts.

Firstly, as Figure 1 illustrates, IGATE takes a text pair $(\boldsymbol{s_1}, \boldsymbol{s_2})$ as input and encodes it by the embedding layer of the PLM, then an embedding matrix is outputted:

$$E = [E_{\text{p}}; E_1; E_2] = \text{Embed}([\boldsymbol{p}, \boldsymbol{s_1}, \boldsymbol{s_2}]) \quad (2)$$

where $E \in \mathbb{R}^{L\times d}$ is the embedding matrix of input, $E_1 \in \mathbb{R}^{L_1\times d}$ and $E_2 \in \mathbb{R}^{L_2\times d}$ are the embedding matrix of $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$, $E_p \in \mathbb{R}^{\ell\times d}$ is the embedding matrix of prompt tokens, L is the sequence length, L_1, L_2 and ℓ are the length of $\boldsymbol{s_1}$, $\boldsymbol{s_2}$ and $\boldsymbol{p}$ separately, d is the embedding size.

Secondly, IGATE extracts semantic information from instances. Considering prompts usually consist of multiple tokens, their information flow can be controlled with different granularity restrictions: token-wise, channel-wise, and element-wise. Meanwhile, the granularity decreases, and the computational complexity and spatial complexity increase gradually. To balance the granularity and computational complexity, IGATE tries to generate channel-wise gating signals: IGATE extracts features through convolutional neural networks (CNNs) with kernels of different sizes. Subsequently, the extracted semantic information is reconstructed and mapped into a vector whose dimension is the same as that of the hidden state of the PLM through the pooling layer as well as the dense layer. Consequently, the corresponding gate weight will be generated for each element of the prompt embedding through the sigmoid function:

$$W_{\text{sem}} = \sigma(\text{Dense}(\text{Pooling}(\text{CNN}(E)))) \quad (3)$$

where $W_{\text{sem}} \in \mathbb{R}^{\ell\times d}$ is a weight matrix.

Finally, IGATE multiplies the prompt embedding and gate weights channel-wise and concatenates the new prompt embedding E'_p with E_1 and E_2:

$$E'_p = \text{Gate}(E_p; E) = W_{\text{sem}} \odot E_p \quad (4)$$

$$E' = [E'_p, E_1, E_2] \quad (5)$$

where $\odot$ demonstrates channel-wise multiplication. Thus far, the vanilla continuous prompt learning method in Equation 1 is converted to IGATE:

$$\mathcal{P}(y|(\boldsymbol{s_1}, \boldsymbol{s_2})) = P(w|\mathcal{M}(E')) \quad (6)$$

3 Experiment

In this section, we detail experimental results on two text matching tasks: MRPC and QQP, as well

	MRPC (F1)	QQP (F1)
Majority	81.2	0.0
LM-BFF(man) (*)(Gao et al., 2021)	74.5 (5.3)	65.5 (5.3)
LM-BFF(auto) (*)(Gao et al., 2021)	76.2 (2.3)	67.0 (3.0)
P-tuning (*)(Liu et al., 2021b)	80.35(1.04)	68.52(2.63)
DART (*)(Zhang et al., 2021)	78.3(4.5)	67.8(3.2)
DCCP (*)(Zhou et al., 2022)	80.3(1.3)	67.9(3.5)
IGATE (avg)	**81.40**(3.99)	**69.05**(1.23)
IGATE (best)	83.61(1.35)	70.52(1.30)

Table 2: Main results on the test sets of MRPC and QQP. (*): Results reported in corresponding papers. Majority: Labeling data with the most frequency label. (avg): In every split averaging performance of multiple experiments. (best): In every split reporting the best performance of multiple experiments.

as three NLI datasets: RTE (Wang et al., 2019b), SNLI (Bowman et al., 2015), and QNLI (Wang et al., 2019b). IGATE has achieved state-of-the-art results on few-shot text matching tasks, exhausting our knowledge. Besides, IGATE can be extended to other prompt learning methods as well.

3.1 Experiments settings

Datasets We evaluate IGATE on MRPC and QQP in few-shot settings and verify the generalizability of IGATE on three NLI datasets: RTE, SNLI, and QNLI. We randomly sample five 16-shot splits with five seeds from every original dataset and follow the same evaluation protocol as Gao et al. (2021). The statistics of all datasets are detailed in A.1.

Setup IGATE is based on P-tuning (Liu et al., 2021b). Thus we compare IGATE with continuous prompt methods, such as P-tuning (Liu et al., 2021b), DART (Zhang et al., 2021), DCCP (Zhou et al., 2022), and a discrete prompt method LM-BFF (Gao et al., 2021). To compare with different baselines fairly, the underlying PLM is RoBERTa-large (Liu et al., 2019). We train the model on five independent training splits and average the F1 score on the test set as the final result. IGATE is implemented using PyTorch. The optimizer of IGATE is AdamW (Loshchilov and Hutter, 2019). We optimize the prompt and also fine-tune the PLM. We use three sizes of kernels with a field of view of 3,5,7 in CNN layers. The settings of all hyper parameters and the scope of the grid search are shown in appendix A.2.

	RTE	SNLI	QNLI
LM-BFF	65.7	64.16	64.16
P-tuning	69.16	69.33	60.90
LM-BFF+gate	63.08	64.38	64.39
P-tuning+gate(IGATE)	**69.17**	**69.81**	**64.97**

Table 3: Results on NLI datasets with RoBERTa-large as the underlying PLM.

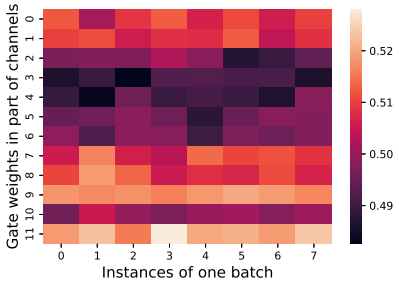

Figure 2: Gate weights in different channels.

3.2 Results

Main results As the main results illustrated in Table 2, IGATE outperforms previous discrete(Gao et al., 2021) and continuous (Zhang et al., 2021; Zhou et al., 2022; Liu et al., 2021b) prompt-based methods. The performance is improved by IGATE for 1.05 points and 0.53 points compared to the previous SOTA model DCCP on MRPC and QQP datasets. Furthermore, IGATE outperforms P-tuning by an average of 1.05 points on MRPC and QQP, which demonstrates that IGATE is effective.

Generalizability and extensibility We conduct experiments on NLI datasets to verify the generalization and extensibility of IGATE. As shown in Table 3, IGATE achieves certain improvements on RTE, SNLI, and QNLI, indicating that IGATE can be applied to other tasks. Furthermore, IGATE restricts the flow of prompt tokens by extracting semantic information from instances as gate structures. We also introduce the gate mechanism in discrete prompt-based methods. Experiments show that IGATE can be extended to the discrete prompt method (Gao et al., 2021).

Analysis We carefully inspect how the gate mechanism affects IGATE. We randomly sampled eight instances and analyzed the gating signal weights generated by different instances on part of the channels. The visualization results which are recorded in Figure 2 illustrates that the gate mechanism can obtain different gate weights for different instances.

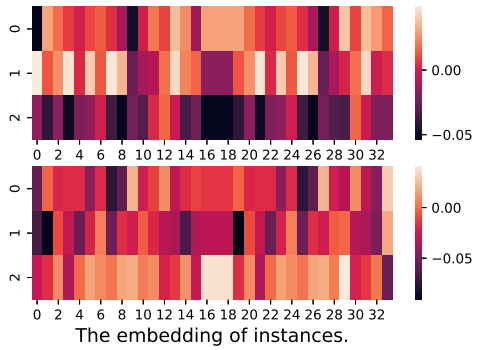

Figure 3: The cosine similarity between prompts and instances before (the upper) and after (the lower) gate.

In addition, we compared the similarity distributions of prompt tokens and tokens of instances with and without the gate mechanism. As Figure 3 shows, the similarity distributions of the prompt and the instance are quite different before and after passing through the gate. This demonstrates that the gate mechanism we introduce can indeed influence the flow of information from prompt tokens.

Ablation study We compare the changes with and without the gate to demonstrate the effectiveness of the gate mechanism in IGATE. The random seed is fixed as ten. We test the performance of IGATE under different conditions on the MRPC task. We remove each part of IGATE separately and record the performance after the removal. The experimental results are shown in Table 4. Moreover, to visually demonstrate the benefit of introducing the gate mechanism, we remove all the extra structures, corresponding to ALL* in Table 4.

Model	MRPC (F1, seed=10)
IGATE	**82.80**($\pm$0.79)
w/o CNN	81.88($\pm$2.65)
w/o sigmoid	81.56($\pm$2.05)
w/o dense layer	81.59($\pm$1.52)
w/o ALL (*)	80.16($\pm$1.83)

Table 4: Ablation results in MRPC. (*): Removing CNN, activation function, and dense layer from IGATE.

4 Conclusion

In this paper, we propose an instance-guided prompt learning method named IGATE. IGATE constructs prompts with the weight matrix which is extracted from the instance. Thus the prompts are restricted by instance semantics in the training and inference process. Experimental results show that IGATE achieves improvement on the text matching task, and can be generalized in NLI tasks. Meanwhile, IGATE can be applied to both discrete and continuous templates.

In the future, we will explore more methods to construct prompts with the semantics of instances.

Limitations

IGATE only works on the lower layers of the PLM and uses a convolutional neural network, thus introducing additional parameters. In fact, the parameter size of the dense layer is proportional to the length of the prompt. This means that in our method, when the prompt length is too long, the parameter scale will be too large, and it is easy to overfit in a few-shot scenario. Above all, the prompt length will be limited after using IGATE.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No.62176024); Beijing Municipal Science & Technology Commission [Grant No.Z181100001018035]; Engineering Research Center of Information Networks, Ministry of Education; the Fundamental Research Funds for the Central Universities (2021XD-A01-1).

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Xiaodong Gu, Kang Min Yoo, and Sang-Woo Lee. 2021. Response generation with context-aware prompt learning. *CoRR*, abs/2111.02643.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5216–5228. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *ArXiv preprint*, abs/2103.10385.

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. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. *CoRR*, abs/2108.13161.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [mask]: Learning vs. learning to recall. In *North American Association for Computational Linguistics (NAACL)*.

Jie Zhou, Le Tian, Houjin Yu, Zhou Xiao, Hui Su, and Jie Zhou. 2022. Dual context-guided continuous prompt tuning for few-shot learning. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 79–84. Association for Computational Linguistics.

A Appendix

In appendix, we detail the experiment settings in five datasets and provide the candidate space of hyper parameters.

A.1 Datasets

We utilize five datasets in experiments: 1) two text-matching datasets: MRPC and QQP. 2) three NLI datasets RTE, SNLI, and QNLI. We randomly sample five splits with different seeds for every dataset. Every split contains the original test set, the 16-shot training set, and the 16-shot development set. Table 5 shows the statistics of these original datasets, and Table 6 shows these statistics in few-shot settings.

Dataset	$\lvert\mathcal{D}_{train}\rvert$	$\lvert\mathcal{D}_{dev}\rvert$	$\lvert\mathcal{D}_{test}\rvert$	#classes
MRPC	3.6k	0.4k	1.7k	2
QQP	363k	40k	390k	2
SNLI	549k	9.8k	9.8k	3
QNLI	104k	5k	5.4k	2
RTE	2.4k	0.27k	3k	2

Table 5: Statistics of datasets

Dataset	$\lvert\mathcal{D}_{train}\rvert$	$\lvert\mathcal{D}_{dev}\rvert$	$\lvert\mathcal{D}_{test}\rvert$	#classes
MRPC	32	32	409	2
QQP	32	32	40430	2
SNLI	48	48	9815	3
QNLI	32	32	5462	2
RTE	32	32	277	2

Table 6: Statistics of datasets in few-shot settings

A.2 Hyperparameters

In our experiments, we divide the embedding to prompt embedding and raw embedding. The prompt embedding layer is set with an independent learning rate. We find that it is hard to choose best parameters for small development sets in training process. So we repeat three times for every set of hyperparameters, and we average the F1/accuracy scores as the experimental results. We set the same scope for every dataset:

- learning rate of PLMs: 1e-5
- batch size: {8, 16}
- prompt learning rate: {1e-5, 5e-5, 1e-4}
- max epochs: {20, 30, 40, 50, 125}

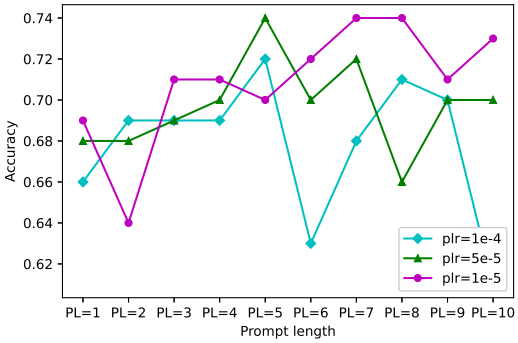

Figure 4: Accuracy on MRPC with different prompt lengths and prompt learning rates.

- prompt length: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
- max sequence length: 128 ,256
- evaluation step: 20, 100

We select hyperparameters by grid search. For example, we record the curves about the prompt length and prompt learning rates, which is illustrated on Figure 4.

We also record the detailed parameters of the IGATE model in Table 7.

layer	paramertes	size
embedding	embedding size	1024
cnn	num of kernel	3
	kernel sizes	3,5,7
	filters (channels)	1024
linear	input size	3 * 1024
	output size	1024

Table 7: Parameters of IGATE model.