Pre-training Extractive Question-Answer Prompts for Few-Shot Chinese Text Classification

Gaojian Ding, Shuang Zheng, and Quanmin Wang(✉)

School of Computer Science, Beijing University of Technology, Beijing, China
djian2021@emails.bjut.edu.cn, {Zshuang,wangqm}@bjut.edu.cn

Abstract. In recent years, pre-training models (PLMs) have made impressive progress, and prompt learning has made few-shot learning achievable. However, traditional prompt learning methods often require manual template design, or performance may be unstable due to the limited data in few-shot tasks. To address these issues, we propose a few-shot text classification method based on multi-task learning. We first unify the multi-task into an extractive question-answering (EQA) format, then train the prompt using task data in the unified format. The prompt cists of modular prompts and a router that indicates their functionality. We then initonsialize the downstream training parameters using the router of a pre-training task similar to the downstream task and employ contrastive learning to improve EQA efficiency.

Keywords: few-shot learning · prompt · multi-task learning · text classification

1 Introduction

Recently, the emergence of pre-trained language models, such as GPT [1] and BERT [2], has dramatically improved the ability of natural language processing to handle downstream tasks. The traditional way to adapt general-purpose PLMs to specific downstream tasks is to fine-tune them by updating all parameters. Therefore, it is necessary to store a modified copy of the full-scale model parameters for each downstream task [3]. So it will be costly when applying the model to downstream tasks [4, 5].

Prompt tuning is a method for adapting PLMs to downstream tasks, consisting of two key engineering techniques: prompt and answer engineering [6]. Prompt engineering works by adding prompts to the input sequence and feeding the new input to the PLM in the pretraining task. In this way, the model will output at the relevant position of the prompt according to the existing knowledge [7] to complete the downstream task. For instance, a common prompt-tuning approach for text classification is to concatenate an input with the prompt "I felt [MASK]" and ask the PLMs to replace it with "happy" or "sad". Discrete prompt tuning, however, has limitations due to PLMs being continuous from an optimization perspective [8]. To overcome this, continuous and deep prompt tuning have been proposed [9], but have their own challenges, such as the weakening

© The Author(s) 2023
K. Hemachandran et al. (Eds.): ICAID 2023, AHIS 9, pp. 318–326, 2023.
https://doi.org/10.2991/978-94-6463-222-4_34

influence of input prompts in intermediate layers and unstable training with additional parameters [10]. Answer engineering involves mapping model-predicted answers to labels, traditionally achieved through manual design. This method requires significant designer input and impacts efficiency [6].

To solve the above problems, we propose a novel deep prompt method to tackle the challenges of downstream tasks with limited data for pre-trained language models. Our method consists of three steps: First, we convert multi-task data into EQA tasks, using all labels as part of the EQA input, which simplifies the construction of the answer project. Second, we use multi-task learning to train deep prompts and employ Sun et al.'s router method to train different router parameters for different tasks in the router structure. This ensures that the training results of multi-task data can quickly adapt to downstream classification tasks, and we can use the routing parameters of a task similar to the downstream task to form the initial prompt when the downstream task starts. In the third step, we leverage contrastive learning to penalize the wrong label output in the few-shot setting, thus enhancing the EQA task's performance.

2 Related Work

2.1 Multi-task Learning

Pre-trained language models are data-driven models that require a large number of labeled training samples, which is usually expensive for NLP tasks that require language knowledge from annotators. To further improve model performance, address data scarcity, and facilitate cost-effective machine learning, researchers have adopted multi-task learning (MTL) for NLP tasks [12]. For example, Liu et al. used an adversarial multi-task learning framework on text classification tasks to alleviate the mutual interference between shared and private latent feature spaces [13]; Vu et al. proposed a migration learning method based on soft prompt, large-scale empirical studies conducted on 26 NLP tasks and significantly improve the performance of prompt tuning in many downstream tasks [14].

2.2 Prompt Engineering

Prompt engineering is a crucial technique that transforms the input forms of downstream tasks into pre-training tasks, bridging the gap between them and PLMs. Prompt engineering can be broadly classified into two types: discrete and continuous. In discrete prompt engineering, prompt templates are manually designed and used directly or for prompt mining to construct new prompts. For instance, the LM-BFF method employs the T5 model to automatically generate prompts [15]. Conversely, continuous prompt engineering is utilized for PLMs without human intervention, where the continuous prompt can be entirely replaced by trainable embeddings. To address the prompt template dependence on prompts in discrete prompt engineering, P-tuning replaces certain text tags with trainable embeddings [8]. In contrast, P-tuning v2 utilizes continuous embeddings as prompts for each layer of input sequences in PLMs to overcome the problem of the lack of generality across scales and tasks in continuous prompts [9].

3 Methodology

In this section, we first introduce a method for converting different tasks to the format of the EQA task, then introduce a method for prompt construction, and finally introduce contrastive learning strategies applied to classification tasks. The frame-work of the model is shown in Fig. 1.

3.1 Unified EQA Format

Few-shot learning usually suffers from limited training data, making it essential to unify the formats of pre-training and downstream tasks via multi-task learning to reduce the gap between them. Previous approaches, such as Gu et al.'s, assume that different tasks only vary in label selection, and address task unification via multiple-choice classification [16]. Conversely, NS et al. and Sun et al. propose unifying all tasks into EQA tasks, by appending labels to the text and adding questions to create a new EQA task [17, 19].

Single-sentence classification tasks often involve numerous classification results, making them ideal for unification into an EQA task format. Our approach divides all tasks into four types: natural language inference (NLI), text classification (TC), extractive question-answering (EQA), and multiple choice question-answering (MCQA). The uniform example format is shown in Table 1.

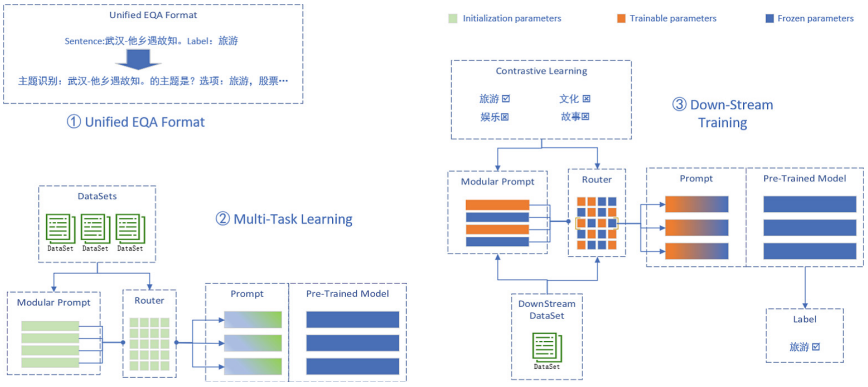

Fig. 1. Model framework

Table 1. Example templates to formulate non-QA tasks into the EQA format

Dataset	Task	Template
Dianping	TC	打分: <s>的评价是?选项: 非常差, 较差, 一般, 较好, 非常好
AFQMC	NLI	意思判别: <s1>与<s2>的意思是?选项: 矛盾, 相似
DogWhistle	MCQA	近义词选择: 与词语<s>最相近的词是?选项: <A1>, <A2> …

3.2 Prompt Construction

To make multi-task learning adaptable to different downstream tasks, we expand a single prompt into a set of modular prompts according to the method of Sun et al. [11] and set the weight parameters of modular prompts for each pre-training task, and the final prompt will be composed by modular prompts and weight parameters.

Specifically, for prompts at each layer l, a set of modular prompts $\left\{p_1^l, \ldots, p_k^l\right\}$ and weight parameters $w = \left\{w_1^l, \ldots, w_k^l\right\} \in \{0, 1\}$ will be predefined, and this parameter composed by weight parameters is called Router. For each task, the final prompt is the weighted mean of this set of modular prompts and weight parameters.

$$P^{(l)} = \frac{1}{K} \sum\nolimits_k^K w_k^l p_k^l \tag{1}$$

During multi-task learning, the modular prompt and weight parameters will be updated. For downstream tasks, we directly select the set of routing parameters similar to the downstream tasks in multi-task learning, use this set of parameters as the downstream training initial parameter, and then prompt tuning it.

3.3 Contrastive Learning Strategies

A sentence can easily be misclassified for single-sentence classification due to some words. For example, For the sentence: "武汉-他乡遇故知", PLM may predict the classification of "旅游" from the word "他乡". It is also possible to mistakenly classify it as a "文化" or "故事" category from the poem "他乡遇故知".

Inspired by contrastive learning [18], we can distinguish between positive and negative predictions, alleviating this problem of confusion. Specifically, we first select k + 1 classes from all possible classes, suppose as $Z = z_1', z_2', \ldots, z_{k+1}'$, the possible predicted results may contain the correct answer z_{cor}, then we will take the remaining k answers as negative answers z_i'; if the result does not contain the correct answer, then we select the top k predicted answers as the negative answer z_i'. For each predicted answer, we have:

$$z_i = \mathrm{Top}_{i_{j,k:j\leq k}} \left(P_{\text{start}}^{(j)} \times P_{\text{end}}^{(k)} \right) \tag{2}$$

where P_{start} and P_{end} represent the probability of each word as the starting position and ending position of the answer, respectively. Then, for each training sample, the contrastive loss function can be described as:

$$L_{SCL} = \frac{1}{K} \sum\nolimits_{i=1}^K max\left(0, \delta - z_{cor} + z_i'\right) \tag{3}$$

where δ ∈ [0,1] is a margin hyperparameter, z_{cor} represents the probability of the correct answer in the predicted answer, and $z_i\prime$ represents the probability of a negative answer in the predicted answer. The final total loss function looks like this:

$$L = L_{MLM} + \lambda L_{SCL} + \gamma \|\Theta\| \tag{4}$$

where L_{MLM} denotes the training objective of token-level MLM. Θ denotes the model parameters. λ, $\gamma \in [0, 1]$ are the balancing hyper-parameter and the regularization hyper-parameter, respectively.

4 Experiments

4.1 Datasets

Multi-task Learning Datasets. As mentioned above, we divide all tasks into 4 types, namely NLI, TC, EQA, and MCQA. We pre-train on 33 Chinese NLP tasks of various types, domains, and sizes with deep modular prompts, and the information on the datasets is shown in Table 2. In multi-task learning, training datasets for different tasks may have different sizes and distributions, which may cause some tasks to be underestimated or overestimated during training, thus affecting the model's performance on the entire task set. We randomly select the task ID and obtain a small batch of data from the corresponding data set so that the model can perform balanced learning among different tasks to alleviate the data imbalance problem.

Downstream Datasets. For the downstream task datasets, we choose 4 classic Chinese classification datasets: Tnews, CSLDCP, EPRSTMT, and IFLYTEK. The details of the data are shown in Table 3. Because the category of EPRSTMT is sentiment analysis of 2 classifications, except for the EQA format of EPRSTMT, which is "<S>的情感是?选项: +标签", the other tasks EQA format is "主题识别: <S> 的主题是?选项: +标签".

Table 2. Multi-task Learning Datasets

Task	Datasets	Size
NLI	AFQMC, Paws, CMNLI, BQ, CHIP-STS, KUAKE-QQR, XNLI, NLPCC-DBQA, Finance-zhidao, Liantong-zhidao, Law-zhidao, Nonghang-zhidao, Touzi-zhidao, Baoxian-zhidao, Dianxin-zhidao, OCNLI	1.98M
TC	CHIP-CTC, FinRe, Fudan-TC, KUAKE-QIC, NLPCC-TC, Amazon, DianPing, DMSC, Online-Shopping, SanWen, THUCNNews	7.96M
QA	DuReader-Checklist, DuReader-Robust, CMRC-2018	24K
MCQA	CCPM, DogWhistle	237K

Table 3. Downstream Datasets

Name	Type	#Class	Test Size
Tnews	ShortTextClassify	15	2010
EPRSTMT	ShortTextClassify	2	610
CSLDCP	LongTextClassify	19	1784
IFLYTEK	LongTextClassify	22	1749

4.2 Experiment Settings

We follow the successful experiment of Sun et al. [11], for multi-task learning, the length of the modular prompt is set to 8, the number of training epochs is set to 2 million times, and the model is trained with a fixed random seed of 42.

In the Downstream Training stage, we utilized the router parameters of the NLPCC-TC task as the initial parameters for the EPRSTMT data set, and for the other three tasks, we opted for the router parameters of Fudan-TC. We ran 1000 rounds of training with random seeds of 4/42/100 for testing. Additionally, we employed contrastive learning for downstream tasks, except for the EPRSTMT dataset. For negative labels, we selected five labels other than the correct label.

For few-shot learning, we perform 1/4/8/16-shot experiments. In the K-shot experiment, we sample K instances of each class from the original training set to form the training set for few-shot learning, and obtain the validation set in the same way. We save the best performing checkpoint on the validation set for testing. In all experiments, we use the accuracy rate as the test metric.

4.3 Backbones and Baselines

We choose Chinese_pretrain_mrc_roberta_wwm_ext_large as our backbone model, which is a retraining model of Roberta-wwm-large based on large-scale MRC data. Because our method converts all tasks into EQA format for pre-training, so we choose this model. We consider (1) Model Tuning, which fine-tunes all parameters of the PTM; (2) PET, through the manual construction of prompt templates, to perform discrete prompt tuning on all parameters of the model [20], (3) P-Tuning, by using continuous prompt embedding as a template, and a small amount of natural language prompts are added as anchor characters to improve the effect, and prompts are tuned for all parameters of the model; and (4) P-Tuning V2, which integrates and adjusts soft prompts at each layer of PTMs, freeze the parameters of the PTMs model, and only perform prompt tuning for the newly added soft prompt as our baselines.

4.4 Main Results

The results of the experiment are shown in Table 4. From the results, we can see that under the setting of few-shot, our method achieves the best results most of the time. Only under the setting of 16-shot in the CSLDCP data set, the results of PET have achieved the best results, but it is enough to prove that our method is effective. We can also see that with the increase of training data, the method of tuning all parameters of the model is gradually narrowing compared with our method. Compared with the P-Tuning v2 method, it also adopts the method of adding soft prompts at each layer, but the initial value is random. It can be found that P-Tuning v2 introduces a new gap, so it also proves that multi-task learning on the added soft prompts before applying them to downstream tasks can help improve the effect of few-shot.

Table 4. Classification results

Shot	Method	Tnews	EPRSTMT	CSLDCP	IFLYTEK
1	FT	25.6 ± 2.3(27.9)	54.2 ± 3.3(57.5)	27.2 ± 1.5(28.7)	23.4 ± 0.4(23.8)
	PET	35.1 ± 1.9(37.0)	64.9 ± 1.1(66.0)	31.6 ± 2.0(33.8)	23.8 ± 2.2(26.0)
	P-Tuning	36.8 ± 3.1(39.9)	51.4 ± 2.2(53.6)	19.2 ± 2.2(21.4)	33.9 ± 1.2(45.1)
	P-Tuning V2	18.1 ± 2.3(20.4)	51.8 ± 0.7(52.5)	21.3 ± 1.9(23.2)	23.2 ± 0.2(23.4)
	PT(Ours)	**47.8 ± 1.3(49.1)**	**67.1 ± 0.2(67.3)**	**49.2 ± 0.2(49.4)**	**40.6 ± 1.3(41.9)**
4	FT	41.0 ± 2.1(42.1)	57.3 ± 2.8(60.1)	34.8 ± 1.7(46.5)	34.3 ± 2.0(36.3)
	PET	45.7 ± 1.1(46.8)	65.1 ± 0.7(65.8)	42.2 ± 1.2(43.4)	41.6 ± 0.8(42.4)
	P-Tuning	43.4 ± 1.2(44.6)	51.6 ± 2.2(53.8)	35.3 ± 2.0(37.3)	40.2 ± 1.3(41.5)
	P-Tuning V2	22.8 ± 3.1(25.9)	54.4 ± 2.1(56.5)	30.8 ± 1.1(31.9)	30.2 ± 0.2(20.4)
	PT(Ours)	**48.1 ± 1.3(49.4)**	**69.1 ± 2.2(71.3)**	**49.2 ± 1.2(50.4)**	**42.0 ± 0.8(42.8)**
8	FT	46.9 ± 2.0(48.9)	63.2 ± 2.9(66.1)	46.2 ± 2.1(48.3)	-
	PET	48.1 ± 2.6(50.7)	66.7 ± 1.2(67.9)	47.7 ± 1.5(49.2)	-
	P-Tuning	45.6 ± 1.1(46.7)	51.9 ± 1.0 (52.9)	46.3 ± 2.7(49.0)	-
	P-Tuning V2	35.1 ± 1.9(37.0)	62.0 ± 2.3(65.3)	41.3 ± 2.2(43.5)	-
	PT(Ours)	**50.4 ± 0.7(51.1)**	**72.1 ± 3.1(75.2)**	**50.1 ± 0.6(50.7)**	-
16	FT	50.3 ± 1.2(51.5)	66.5 ± 2.7(69.2)	51.9 ± 1.2(53.1)	-
	PET	51.1 ± 1.3(54.4)	72.2 ± 0.3(72.5)	**55.2 ± 1.1(56.3)**	-
	P-Tuning	51.8 ± 1.1(52.9)	53.0 ± 1.5(54.5)	53.5 ± 1.0(54.5)	-
	P-Tuning V2	46.6 ± 2.2(48.8)	67.7 ± 3.4(71.1)	40.2 ± 2.2(42.4)	-
	PT(Ours)	**52.1 ± 2.1(54.3)**	**81.2 ± 2.1(83.3)**	51.6 ± 1.0(52.6)	-

5 Conclusion

In this paper, we propose a method to tackle few-shot text classification by unifying diverse datasets in an EQA format and employing MTL-based pre-training with tailored prompt routing initialization. Additionally, we use contrastive learning to narrow the gap with incorrect answers during downstream tasks. Our approach outperforms baseline models in the few-shot setting. Nonetheless, our method has certain limitations, such as high initial training costs (about 192 h in a 3090 environment) and a comparison loss function that does not account for word similarity. These challenges will guide our future work.

References

1. Bubeck S, Chandrasekaran V, Eldan R, et al. Sparks of artificial general intelligence: Early experiments with gpt-4[J]. arXiv preprint arXiv:2303.12712, 2023.
2. Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
3. Gururangan S, Marasović A, Swayamdipta S, et al. Don't stop pretraining: Adapt language models to domains and tasks[J]. arXiv preprint arXiv:2004.10964, 2020.
4. Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning[J]. arXiv preprint arXiv:2104.08691, 2021.
5. Chen S, Hou Y, Cui Y, et al. Recall and Learn: Fine-tuning Deep Pretrained Language Models with Less Forgetting[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020: 7870-7881.
6. Liu P, Yuan W, Fu J, et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing[J]. ACM Computing Surveys, 2023, 55(9): 1-35.
7. Petroni F, Rocktäschel T, Riedel S, et al. Language Models as Knowledge Bases?[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 2463–2473.
8. Liu X, Zheng Y, Du Z, et al. GPT understands, too[J]. arXiv preprint arXiv:2103.10385, 2021.
9. Liu X, Ji K, Fu Y, et al. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks[J]. arXiv preprint arXiv:2110.07602, 2021.
10. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
11. Sun T, He Z, Zhu Q, et al. Multi-Task Pre-Training of Modular Prompt for Few-shot Learning[J]. arXiv preprint arXiv:2210.07565, 2022.
12. Chen S, Zhang Y, Yang Q. Multi-task learning in natural language processing: An overview[J]. arXiv preprint arXiv:2109.09138, 2021.
13. Liu P, Qiu X, Huang X. Adversarial multi-task learning for text classification[J]. arXiv preprint arXiv:1704.05742, 2017.
14. Vu T, Lester B, Constant N, et al. Spot: Better frozen model adaptation through soft prompt transfer[J]. arXiv preprint arXiv:2110.07904, 2021.
15. Gao T, Fisch A, Chen D. Making pre-trained language models better few-shot learners[J]. arXiv preprint arXiv:2012.15723, 2020.
16. Gu Y, Han X, Liu Z, et al. PPT: Pre-trained Prompt Tuning for Few-shot Learning[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2022: 8410–8423.
17. Sun T, Shao Y, Li X, et al. Learning sparse sharing architectures for multiple tasks[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(05): 8936–8943.
18. Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations[C]//International conference on machine learning. PMLR, 2020: 1597–1607.
19. Keskar NS, McCann B, Xiong C, et al. Unifying question answering, text classification, and regression via span extraction[J]. arXiv preprint arXiv:1904.09286, 2019.
20. Schick T, Schütze H. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference[C]//Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. 2021: 255–269.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.