Learning a Better Initialization for Soft Prompts via Meta-Learning

Yukun Huang* Columbia University yh3310@columbia.edu

Kun Qian* Columbia University kq2157@columbia.edu

Zhou Yu Columbia University zy2461@columbia.edu

Abstract

Prompt tuning (PT) is an effective approach to adapting pre-trained language models to downstream tasks. However, prompt tuning doesn't perform well under few-shot settings due to the poor initialization. So pre-trained prompt tuning (PPT) (Gu et al., 2022) is proposed to adapt prompt tuning to few-shot settings by initializing prompts with source data. We propose **Meta**-learned **P**rompt **T**uning (**MetaPT**) to further improve PPT's few-shot learning performance by considering latent structure within the source data. Specifically, we introduce the framework by first clustering source data into different meta-training tasks in an unsupervised manner. Then we leverage these tasks to meta-train prompts with a meta-learning algorithm. Such a process enables prompts to learn a better initialization by discovering commonalities among these meta-training tasks. We evaluate our method on seven downstream sentiment tasks. The results demonstrate that our MetaPT achieves better performance and stability than the state-of-the-art method.

1 Introduction

Pre-trained language models (PLMs) have demonstrated outstanding performances in various downstream NLP tasks (Devlin et al., 2019, Kale and Rastogi, 2020, Brown et al., 2020). **Full model tuning** (FT) adapts PLMs to downstream tasks by introducing task-specific training objects and fine tuning all parameters of PLMs. **Prompt tuning** (PT) (Lester et al., 2021) is an efficient alternative to FT by only tuning a small number of parameters. PT adds a series of tunable tokens (soft prompts) at the beginning of the sequence to modulate the behavior of the language model. When adapting pre-trained language models to downstream tasks, PT freezes all parameters of pre-trained language models and only trains the soft prompts. PT achieves comparable performance to FT with sufficient data. But it performs poorly under few-shot settings due to its sensitivity to the initialization of soft prompts.

To adapt PT to few-shot settings, **pre-trained prompt tuning** (PPT) (Gu et al., 2022) pre-trains soft prompts with a self-supervised source task and then apply pre-trained prompts to few-shot downstream tasks. PPT generally groups all text classification tasks into three formats and designs a self-supervised source task for each format to pre-train prompts. PPT demonstrates its effectiveness when using large-scale PLMs. However, PPT mixes all source data points together and ignores the latent structure among them. PPT updates prompt parameters at every data point, it learns more about the specific feature of each data point rather than the general features of the entire task. As a result, PPT retains too much redundant information only relevant to the source task in the initialization of soft prompts, which consequently impedes model performance on downstream tasks.

To further improve the few-shot adaption capability of prompt tuning, we incorporate meta-learning into prompt tuning. We first propose to use unsupervised methods to create meta-training tasks for prompts and then adopt a model-agnostic meta-learning method to meta-train prompts. By our unsupervised clustering method, the latent structure of source data is represented by the distribution of meta-training tasks. Through meta-learning, general features are incorporated into the initialization of the soft prompts. Our meta-learned prompts achieve faster and more stable adaptation to downstream tasks. We named our method **Meta**-learned **P**rompt **T**uning (**MetaPT**). Our experimental results show that MetaPT outperforms full-model tuning and pre-trained prompt tuning on the T5 model (Kale and Rastogi, 2020).

2 Related Work

Prompts There are two types of prompts that can probe the knowledge in PLMs without updating

*Equal Contribution

Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers), pages 67–75
November 1–4, 2023.

the parameters of PLMs: hard prompts and soft prompts. Hard prompts (Brown et al., 2020) are human-designed discrete tokens while soft prompts are continuous embeddings of language models. Soft prompts methods train efficient parameters to perform prompting directly into the continuous embedding space of the model to get better representation ability and avoid involving human efforts. These efficient parameters can be prepended to each layer in the encoder stack (Li and Liang, 2021), input embedding (Lester et al., 2021), or both (Liu et al., 2021). Though the above soft prompts methods perform well with sufficient training data, they all become much worse under few-shot learning settings. Pre-trained prompt tuning (Gu et al., 2022) is the current state-of-the-art soft prompts method under few-shot settings.

Meta-Learning Meta-Learning, also known as learning to learn, is famous for its effectiveness to extract domain-invariant features (Sahoo et al., 2018) and enforces models to adapt to downstream tasks (Finn et al., 2017). Model-Agnostic Meta-Learning (MAML) proposed by Finn et al. (2017) is a popular optimization-based meta-learning algorithm, which is adopted in various NLP tasks (e.g. Qian et al., 2021, Gu et al., 2018, Yin, 2020, Qian and Yu, 2019, Dou et al., 2019). Following MAML, works like REPTILE (Nichol et al., 2018), MetaOPT (Lee et al., 2019) and TAML (Jamal et al., 2019) etc. are proposed to further improve the model's learning capability. In this work, we focus more on exploring how soft prompts benefit from meta-learning. Therefore, we adopt the most widely used MAML algorithm.

3 Background: Prompt Tuning

Prompt tuning modifies the embeddings of soft prompts prepended to the input sequence to adapt PLMs to downstream tasks. Specifically, let (x, y) be a sample from a downstream classification task $\mathcal{T}^{down}$, where x is the input text while y is its corresponding label. We first map the label y into a concrete token z in the vocabulary of the PLM. Next, we use $H(\cdot)$ to fit input text x to a hard prompt template. Take sentiment classification as an example, H("I like this movie") ="I like this movie. It was $\langle$X$\rangle$ ", where $\langle$X$\rangle$ is the mask token. Then, we prepend soft prompts $\boldsymbol{P}$ to the beginning of input sequence $[\boldsymbol{P}; H(x)]$. Finally, we freeze the parameters of the PLM and optimize the following log-likelihood objective:

$$\underset{\boldsymbol{P}}{\operatorname{argmax}} \sum \log p(\langle \mathrm{X} \rangle = z | [\boldsymbol{P}; H(x)]; \boldsymbol{P})$$

4 Meta-learned Prompt Tuning

In this section, we describe our self-supervised model training pipeline. We first gather source data. Then, we utilize an unsupervised method to cluster source data into different groups as meta-training tasks. Finally, we use Prompt-MAML algorithm to train and find a good initialization for soft prompts.

Our algorithm helps the model adapt faster to downstream tasks in two ways. On one hand, the meta-learning method simulates the adaptation step during training, which provides an easily-optimized parameter initialization. On the other hand, training across different clusters allows the model to focus more on the task's general features rather than domain-specific features.

4.1 Constructing Source Data

We construct source data by creating pseudo labels for sentences from open-domain corpus. For sentence classification, we first train a small model based on an existing dataset which shares similar label space to the downstream datasets. Then we use that model to annotate pseudo labels for the sentences in a large corpus.

4.2 Designing Meta-training Tasks

After constructing source data, we use K-means to group the data into different clusters as meta-training tasks. We first implement sentence-BERT (Reimers and Gurevych, 2019) to derive semantically meaningful sentence embeddings from source data samples. Then we apply unsupervised K-means to cluster source data into different classes according to their sentence embeddings.

K-means clustering group samples with similar sentence embeddings into the same task and reveal the latent structure within source data. Based on that structure, prompts could learn to incorporate some common internal features to the initialization through meta-learning. With such general information encoded in the initialization of prompts, the model can achieve great performance with limited training data from downstream tasks.

4.3 Prompt-MAML Algorithm

After we get a set of meta tasks $\mathcal{T}$ obtained via an unsupervised method, we utilize MAML to learn

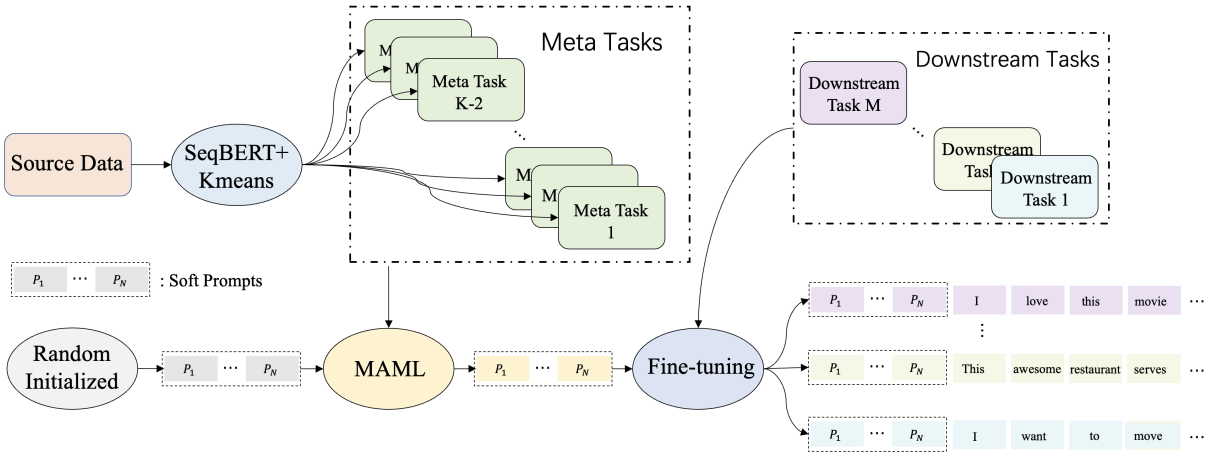


Figure 1: Pipelines of meta-learned prompt tuning. First, we prepare source data used for meta-learning. Second, we cluster source data into different groups as meta-training tasks (aka meta tasks). Then, we train prompts with model-agnostic meta-learning algorithm. Finally, we evaluate meta-learned prompts on downstream tasks.

general features among these meta tasks. We first randomly initialize the parameter of soft prompts $\boldsymbol{P}$. For each meta task $\mathcal{T}_i$, m training samples are sampled from that task. Taking in m samples, the model output $f_{\boldsymbol{P}}$. Then we calculate the average loss $\mathcal{L}_{\mathcal{T}_i}(f_{\boldsymbol{P}})$ of these m samples and temporarily update soft prompts with gradient descent, where α is the learning rate for the inner loop.

$$\boldsymbol{P_i'} = \boldsymbol{P} - \alpha \nabla_{\boldsymbol{P}} \mathcal{L}_{\mathcal{T}_i}(f_{\boldsymbol{P}}) \tag{1}$$

After optimizing the prompts, we sampled another m samples and calculated the loss with the updated prompts. We add loss for $\mathcal{T}_i$ to total loss and repeat the same process for other meta tasks until we go over all the meta tasks. Finally, we update the prompts by minimizing the final total loss, where β is the learning rate for the outer loop.

$$\boldsymbol{P} \leftarrow \boldsymbol{P} - \beta \nabla_{\boldsymbol{P}} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{T_i}(f_{\boldsymbol{P_i'}}) \tag{2}$$

This is a complete process of one-step updates for prompts. We keep optimizing the prompts until the validation accuracy of meta tasks stop growing. During meta-training, we simulate the adaptation step and therefore provide an easily-optimized soft prompts initialization. During the evaluation, the meta-learned prompts will be further tuned on downstream tasks and then make predictions.

5 Experiments

5.1 Setup

Our experiments are built on the T5-base model from HuggingFace (Wolf et al., 2020).

Downstream Datasets We focus on the sentiment classification tasks. Specifically, the downstream datasets include SST-5 (Socher et al., 2013), SST-2 (Socher et al., 2013), Amazon-5 (Zhang et al., 2015), Amazon-2 (Zhang et al., 2015), Sentihood (Saeidi et al., 2016), and SemEval-2016 (Pontiki et al., 2016). SemEval-2016 has two tasks in different domains: SemEval$_r$(restaurant) and SemEval$_l$(laptop). Detailed information on these datasets could be found in Appendix C. We randomly select 40 samples from the original dataset for both few-shot training and validation.

Source Data we first train a RoBERTa-base model on Yelp5. Then we randomly sample 10GB of data from OpenWebText (Gokaslan and Cohen, 2019) and apply the trained RoBERTa model to annotate labels for the sampled data. We only keep data samples with high confidence and throw away the samples which the model is unsure about. After balancing pseudo data, we get 1,000,000 balanced training samples with open domains.

See Appendix A for detailed hyperparameters.

5.2 Main Results

As shown in Table 1, we mainly compare the performance of full-model tuning (FT), pre-trained prompt tuning (PPT) and meta-learned prompt tuning (MetaPT) on different sentiment classification tasks. We also include results of MetaPT$_{(Y)}$ in the table, which is directly meta-trained on Yelp5.

First, MetaPT consistently achieves higher accuracy than PPT (7/7 tasks) and FT(6/7 tasks). MetaPT also shows better stability facing different training samples. Second, MetaPT$_{(Y)}$ also out-

Model	Methods	SST5	SST2	Amazon5	Amazon2	Sentihood	SemEval$_r$	SemEval$_l$
T5-base	FT	$43.57_{\pm 2.56}$	$88.27_{\pm 1.03}$	$48.40_{\pm 1.48}$	$92.35_{\pm 0.68}$	$\mathbf{82.11_{\pm 1.30}}$	$71.01_{\pm 1.16}$	$62.48_{\pm 3.23}$
	PPT	$42.90_{\pm 1.08}$	$87.42_{\pm 1.15}$	$51.15_{\pm 1.56}$	$93.28_{\pm 0.21}$	$80.06_{\pm 3.31}$	$62.04_{\pm 3.34}$	$56.37_{\pm 4.11}$
	MetaPT	$45.26_{\pm 0.39}$	$\mathbf{89.47_{\pm 0.12}}$	$55.47_{\pm 0.34}$	$94.43_{\pm 0.08}$	$80.38_{\pm 0.46}$	$76.93_{\pm 1.19}$	$70.86_{\pm 1.95}$
	MetaPT$_{(Y)}$	$\mathbf{46.24_{\pm 0.42}}$	$87.26_{\pm 0.73}$	$\mathbf{58.73_{\pm 0.13}}$	$\mathbf{95.39_{\pm 0.03}}$	$78.27_{\pm 1.17}$	$\mathbf{80.72_{\pm 0.60}}$	$\mathbf{72.32_{\pm 0.66}}$
T5-large	MetaPT	$47.31_{\pm 0.21}$	$89.61_{\pm 0.09}$	$55.76_{\pm 0.15}$	$95.23_{\pm 0.03}$	$85.78_{\pm 0.10}$	$85.96_{\pm 0.08}$	$77.49_{\pm 0.11}$

Table 1: Classification accuracy results on seven downstream tasks. FT denotes full-model tuning. PPT denotes pre-trained prompt tuning. MetaPT denotes meta-learned prompt tuning. MetaPT is meta-trained on pseudo-labeled data while MetaPT$_{(Y)}$ is meta-trained on Yelp5 directly. Our methods outperform the two other methods for most of the datasets. MetaPT not only achieve higher accuracy than PPT consistently, but also have a more stable performance with lower variance.

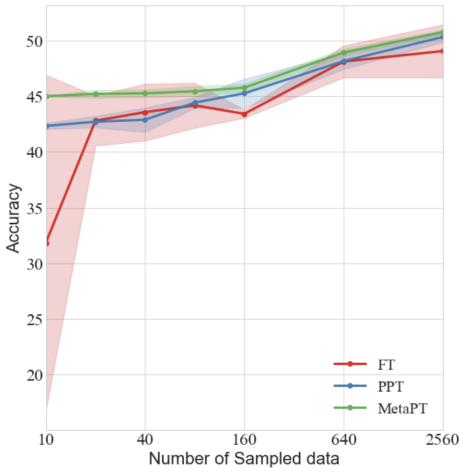
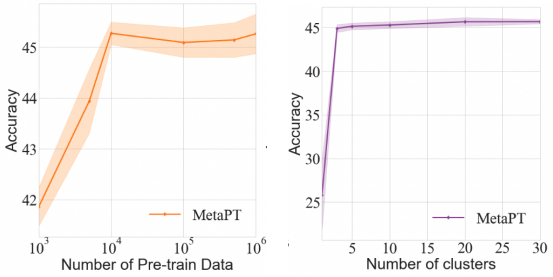


Figure 2: Performance comparison among FT, PPT, MPT on SST-5 as the number of training samples increases from 10 to 2,560



(a) source data size (b) number of clusters

Figure 3: Analysis about MetaPT. (a) The performance of MetaPT varies when the number of pretraining samples changes from 1,000 to 3,000,000. (b)The performance of MetaPT varies when the number of meta tasks varies from 3 to 30

performs PPT and FT. And it even achieves better results than MetaPT on five tasks. Even though it only trained on the restaurant domain, it can still be generalized to other domains. This suggests that our method can extract general features from data. Finally, as we increased the size of the pre-trained language model from base to large, the performances of MetaPT become consistently better on all seven tasks.

In addition, as shown in Figure 2, when the number of training samples of the downstream task grows from 10 to 2,560, MetaPT is consistently better than PPT and FT, while PPT also has a small advantage over FT. All three methods will converge to similar performance with sufficient training data.

5.3 Ablation Study

Scale of Source data As Figure 3(a) shows, when we increase the number of source data points from 1,000 to 10,000, the accuracy grows rapidly. After 10,000 training samples, the performance does not change much as the number of training samples increases. This result suggests that more source data samples do not necessarily lead to better performance in our method. When the size of source data reaches the level of 10,000, it is enough for our model to get acceptable performance.

Number of clusters We examine the cluster number from 3-30 for K-means and compare performance. As shown in Figure 3(b), the accuracy grows rapidly at first as the cluster number increases but later it converges. Considering both effectiveness and efficiency, MetaPT is able to achieve promising results when k=10. We also visualize the result of K-means clustering and compare K-means with other clustering methods. See Appendix B for more details.

6 Conclusions

In this paper, we present the meta-learned prompt tuning framework. Specifically, we propose to cluster source data into different groups to create aux-

iliary tasks for meta-learning, and then meta-train prompts with the Model-Agnostic Meta-Learning method. Our method tunes only 0.02% parameters but improves the accuracy by 3.8% compared with full model tuning on seven downstream tasks.

7 Limitations

In this work, we mainly focus on sentiment classification tasks. Our method could be further explored on other natural language understanding tasks like sentence-pair classification and multiple-choice classification. Besides, our experiments are conducted on T5-base and T5-large models in this work. There are still other available larger pre-trained language models like T5-xl and T5-xxl. The performance of our method on larger language models needs to be further investigated. In the future, we plan to apply our method to these two larger pre-trained language models. We also plan to extend our evaluation tasks from sentiment classification to other general natural language processing tasks, e.g. sentence pair classification, to explore the generalizability of our method.

8 Ethical Considerations

We present a parameter-efficient method to adapt the large language models (LLMs) to few-shot learning tasks, which makes LLMs accessible to more people. However, as LLMs become more accessible, they are more likely to be used maliciously. Our method might open up the potential for scams and fraud on a large scale. For example, a chatbot can be trained to extract sensitive information from users by tuning a few parameters with only a few labeled samples. As a result, a malicious chatbot can be easily trained to deceive users and extract their private information. Therefore, our method can only be put into real life when malicious goals of LLMs can be detected and the user can be warned about potential dangers.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1192–1197, Hong Kong, China. Association for Computational Linguistics.

Chelsea Finn, P. Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423, Dublin, Ireland. Association for Computational Linguistics.

Muhammad Abdullah Jamal, Guo-Jun Qi, and Mubarak Shah. 2019. Task agnostic meta-learning for few-shot learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11711–11719.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. 2019. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10657–10665.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

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

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *ArXiv*, abs/1803.02999.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Kun Qian, Wei Wei, and Zhou Yu. 2021. A student-teacher architecture for dialog domain adaptation under the meta-learning setting. In *AAAI*.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2639–2649, Florence, Italy. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. 2016. SentiHood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1546–1556, Osaka, Japan. The COLING 2016 Organizing Committee.

Doyen Sahoo, Hung Le, Chenghao Liu, and Steven CH Hoi. 2018. Meta-learning with domain adaptation for few-shot learning under domain shift.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. 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.

Wenpeng Yin. 2020. Meta-learning for few-shot natural language processing: A survey. *arXiv preprint arXiv:2007.09604*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *ArXiv*, abs/1509.01626.

A Training Settings

Hyperparameters Following Lester et al. (2021), we set the soft-prompt as 100 tunable tokens for all methods. We provide detailed training settings used for full-model tuning (FT), pre-trained prompt tuning (PPT) and meta-learned prompt tuning (MetaPT). Instead of following Gu et al. (2022), we find another set of hyperparameters. Both FT and PPT achieve better performances on T5-base model than results reported in Gu et al. (2022). We run FT, PPT and MetaPT on T5-base (220M). For MetaPT, we also run experiments on T5-large (770M). All the models in our work could be fit in a single NVIDIA RTX A6000. For few-shot settings, we randomly select 40 training samples and 40 validation samples from the data five times with random seed in [1,2,3,4,5]. We report the averaged accuracy as well as the standard deviation.

A.1 Full-model Tuning

We implement AdamW as the optimizer. We search the learning rate in [3e-3, 3e-4, 8e5, 3e-5, 3e-6], max epochs in [50, 100, 200, 300] and patience in [1,3,5,8]. Then we choose the set of hyperparameters with best accuracy on validation set. We apply a linear scheduler with 20 warm up steps and set the learning rate to 0.00003. We set batch size to 4, max epochs to 200. We evaluate results on validation set every epoch and and set the patience for early stopping to 5. Full-model Tuning would take 1 hour in average.

A.2 Pre-trained Prompt Tuning

We apply source data created in section 5 as pre-training data for PPT. During the pre-training phase, we implement AdamW as the optimizer. We apply the linear scheduler with 20 warm-up steps and set the learning rate to 0.003. We search the evaluation steps in [20,000, 10,000, 5,000] and patience in [1,3,5]. Then we choose the best set of hyperparameters for the pre-training phase. We set the batch size to 4 and the max epoch to 5 (1,250,000 max steps). We evaluate prompts on the validation set every 20,000 steps after the first epoch and set the patience of the early stop to 5. Pre-training phase would take around 36 hours.

A.3 Meta-learned Prompt Tuning

During the meta-learning phase, we use AdamW as the optimizer for the outer loop. We search the learning rate α in [0.8, 0.3, 0.08, 0.03, 0.008] and search the learning rate β in [0.3, 0.08, 0.03, 0.025, 0.008]. Then we choose the best set of hyperparameters for the meta-learning phase. We set the learning rate α to 0.08, learning rate β to 0.025, batch size to 4, early stop patience to 6, and the max updating step of MAML to 20,000. We evaluate the prompts every 500 steps. The meta-learning phase would take around 12 hours. We also run MetaPT on T5-large. We adopt a similar setting as T5-base. The meta-learning phase would take around 24 hours.

A.4 Downstream Prompt Tuning

After pre-training or meta-training the prompts, we adopt the same setting for prompt tuning on downstream tasks. We use the AdamW optimizer with a learning rate of 0.003. We apply the linear scheduler with 20 warm-up steps. We set the batch size to 4 and the gradient accumulation steps to 8. We set the max epoch to 200 and the patience of the early stop to 6.

B Clustering

B.1 Clustering Methods

We compare four different methods of clustering to get meta-training tasks. They are K-means clustering (splitting data by K-means clustering based on sentence embeddings), LDA clustering (splitting data by Latent Dirichlet Allocation topic modeling), random clustering (splitting the data randomly), and label clustering (splitting data according to their pseudo labels). From the result shown in Table 2, we notice that K-means clustering is the most effective and LDA is next to it. When we cluster randomly or according to their labels, the performance of MetaPT degrades to the same level as PPT.

Methods	Accuracy
K-means	46.24±0.42
LDA	44.10±0.83
random	42.95±1.05
label	42.84±0.76
PPT	42.89±1.08

Table 2: Performance of different clustering methods on SST5. "K-means" denotes and "LDA" denotes . "Random" denotes the method which randomly splits the total source data into different groups. "Label" denotes the method which clusters the data samples with the same label into the same group. Pre-trained Prompt Tuning(PPT) plays the role as a baseline.

B.2 Visualization

We use TSNE to visualize the clustering results for K-means when the cluster number equals 3, 6, 10 in Figure 4. From the t-SNE of our K-means clusters, we could see that data is well grouped into different clusters according to their sentence embeddings. After we reduce the sentence features of different samples to two-dimensionality, different samples in the same clusters are close to each other and are distinguishable from samples in other clusters, which demonstrates that meta tasks derived from K-means indeed contain useful common latent features.

We conduct manual inspection on the resulting clusters of the source data (k=10). We find a few human-interpretable structures. For example, some are more related to food, some have more numbers, dates, or symbols, and some include more short sentences. The pattern of other clusters is less interpretable.

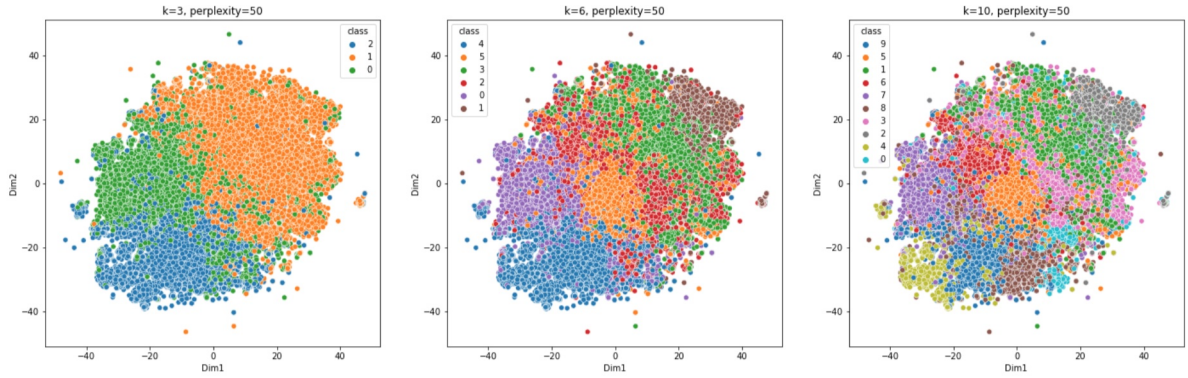


Figure 4: T-SNE of K-means meta tasks clustering results. Cluster number K equals to 3, 6, 10 respectively

C Dataset Examples

Here we provide detailed information and examples for all the datasets we used. Source dataset includes Yelp5 (Zhang et al., 2015). The downstream datasets include SST-5 (Socher et al., 2013), SST-2 (Socher et al., 2013), Amazon-5 (Zhang et al., 2015), Amazon-2 (Zhang et al., 2015), Sentihood (Saeidi et al., 2016), and SemEval-2016 (Pontiki et al., 2016). SemEval-2016 has two tasks in different domains: restaurant and laptop. These two tasks are denoted by SemEval_r and SemEval_l respectively. Domains, number of classes and examples of all datasets are shown in Table 3. All datasets are in English.

Dataset	Domain	classes	Example
Yelp-5	restaurant	5	"dr. goldberg offers everything i look for in a general practitioner. he's nice and easy to talk to without being patronizing; he's always on time in seeing his patients; he's affiliated with a top-notch hospital (nyu) which my parents have explained to me is very important in case something happens and you need surgery; and you can get referrals to see specialists without having to see him first. really, what more do you need? i'm sitting here trying to think of any complaints i have about him, but i'm really drawing a blank." *positive++*
SST-5	movie	5	"unlike the speedy wham-bam effect of most hollywood offerings , character development – and more importantly , character empathy – is at the heart of italian for beginners" *positive++*
SST-2	movie	2	"jason x is positively anti-darwinian : nine sequels and 400 years later , the teens are none the wiser and jason still kills on auto-pilot " *negative*
Amazon-5	product	5	"nice screen for a nice price but..... i compared a few different flat panels with review before i narrowed down my pick, which ended up with the sylvania as over well liked. the picture got great reviews which yes it does have a good picture to look at but there are other important qualities you enjoy that makes viewing tv all the better. for example: sound... how was that forgotten?in this flat panel, it was. what a disappointment. if this is consider stereo than why does it sound like its coming from a tin can with no base at all. then too boot, if you play the dvd, the sound drops and you have to really turn up the volume to hear.i want the whole package deal: space saving, great picture, and good sound. i want to enjoy the whole experience of watching and listening. how about you?" *positive*
Amazon-2	product	2	"not an ultimate guide. firstly,i enjoyed the format and tone of the book (how the author addressed the reader). however, i did not feel that she imparted any insider secrets that the book promised to reveal. if you are just starting to research law school, and do not know all the requirements of admission, then this book may be a tremendous help. if you have done your homework and are looking for an edge when it comes to admissions, i recommend some more topic-specific books. for example, books on how to write your personal statment, books geared specifically towards lsat preparation (powerscore books were the most helpful for me), and there are some websites with great advice geared towards aiding the individuals whom you are asking to write letters of recommendation. yet, for those new to the entire affair, this book can definitely clarify the requirements for you." *negative*
Sentihood	neighbor	2	"a friend of mine lived in location1 and she liked it, though other people have told me it's a bit rough" *negative*
SemEval_r	restaurant	3	"if you've ever been along the river in weehawken you have an idea of the top of view the chart house has to offer" *positive*
SemEval_l	labtop	3	"so if anyones looking to buy a computer or laptop you should stay far far away from any that have the name toshiba on it" *negative*

Table 3: Detailed information about sentiment datasets, including domain, number of classes and a concrete example