MetaPrompting: Learning to Learn Better Prompts

Yutai Hou[*], **Hongyuan Dong**[*], **Xinghao Wang**, **Bohan Li**, **Wanxiang Che**[†]

Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{ythou, hydong, xhwang, bhli, car}@ir.hit.edu.cn

Abstract

Prompting method is regarded as one of the crucial progress for few-shot nature language processing. Recent research on prompting moves from discrete tokens based "hard prompts" to continuous "soft prompts", which employ learnable vectors as pseudo prompt tokens and achieve better performance. Though showing promising prospects, these soft-prompting methods are observed to rely heavily on good initialization to take effect. Unfortunately, obtaining a perfect initialization for soft prompts requires understanding of inner language models working and elaborate design, which is no easy task and has to restart from scratch for each new task. To remedy this, we propose a generalized soft prompting method called MetaPrompting, which adopts the well-recognized model-agnostic meta-learning algorithm to automatically find better prompt initialization that facilitates fast adaptation to new prompting tasks. Extensive experiments show MetaPrompting tackles soft prompt initialization problem and brings significant improvement on four different datasets (over 6 points improvement in accuracy for 1-shot setting), achieving new state-of-the-art performance.

1 Introduction

Enabling models to learn from a few labeled examples, i.e., Few-Shot Learning (FSL), is one of the key steps toward more human-like artificial intelligence. Recently, taking advantage of large-scale Pretrained Language Models (PLM) (Brown et al., 2020), prompting-based methods achieve impressive results for few-shot learning of Natural Language Processing (NLP) (Gao et al., 2021; Liu et al., 2021a; Zhao et al., 2021).

Prompting-based methods insert a piece of text, i.e. prompts, to the input examples, so that the few-shot task can be formulated as a (masked) language
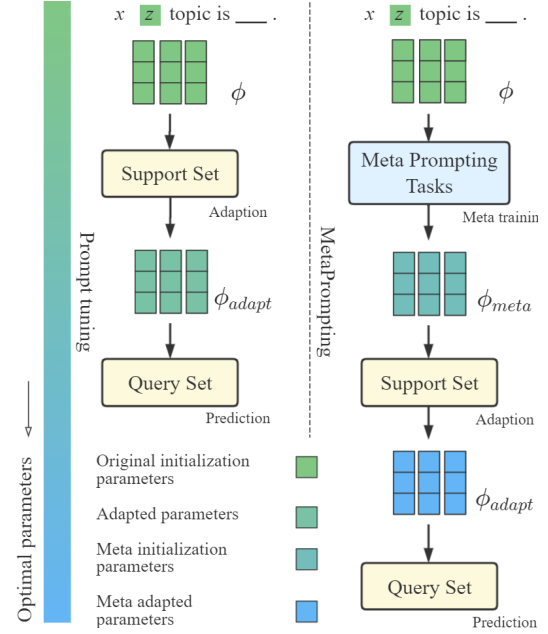

Figure 1: Comparison between conventional soft-prompting method (left) and proposed MetaPrompting (right). x denotes the query sentence, and z is learnable pseudo tokens in soft prompts. ϕ represents all trainable parameters. MetaPrompting exploits optimization-based meta-learning to find an initialization ϕ_{meta} that facilitates better and faster adaptation to new tasks.

modeling problem. For example, say we want to classify the sentiment of the book review "I will never read it again.", we can append a prompt "It was" to the sentence, getting "I will never read it again. It was". It is natural to expect a higher probability from the PLM to generate "terrible" than "great" then. Such converting bridges the gap between pre-training and target tasks. Consequently, it has better transferability and less dependence on target task data.

The performance of prompting methods is found to be greatly affected by the design of prompts (Gao et al., 2021). That is, a good prompt makes significant difference. Early attempts take manually-designed prompts or search prompts automatically.

---

[*] Equal contribution.
[†] Email corresponding

Schick et al. (2020) and (Schick and Schütze, 2021) explore to automatically identify label words. In pursuit of better performance compared to hand-picked prompts, Gao et al. (2021) proposes LM-BFF to search both prompt templates and label words. AutoPrompt (Shin et al., 2020) leverages gradient-based searching to find out the best prompts. These prompts consist of discrete tokens, which limits the prompt search space. To further liberate the potential of prompts, recent works employ learnable vectors as prompt content and learn optimal prompts in continuous space, which is so-called "soft prompts" (Liu et al., 2021c; Li and Liang, 2021). Since they no longer require prompts to be composed of real words, these methods greatly expand the possibilities of prompts and thus achieve better performance (Liu et al., 2021b).

However, despite the promising prospects of soft prompts, learning a good prompt is still far from trivial. Because soft-prompts search for optimal solutions in an infinite continuous space, the choice of the starting point for the search (i.e., prompt initialization) becomes crucial. Soft-prompt is observed to be more sensitive to different initialization than discrete prompts in low data setting (Li and Liang, 2021; Liu et al., 2021b). Unfortunately, creating a perfect prompt initialization requires both understanding of LMs' inner workings and trial-and-error. Lester et al. (2021) initialize soft prompt with the token embeddings of hand-crafted prompt directly. Zhong et al. (2021b) search discrete tokens as better initialization, which shows better performance. What's worse is that these initializations are task-bounded. Every time we confront a new task, the costly process of initialization design has to start from scratch.

In this paper, to tackle the above issues, we let loose the prompt design of a specific task, but instead focus on obtaining a task general prompt initialization that facilitates faster and better adaptation to new prompting tasks. Recently proposed optimization-based meta-learning algorithms, such as MAML (Finn et al., 2017) and Reptile (Nichol et al., 2018), achieve better adaption by learning a parameter initialization. Following their essence, we tackle soft prompt initialization problem by proposing MetaPrompting, which is a generalized soft prompting method powered by meta-learning algorithms. MetaPrompting learns general meta-knowledge from source domain tasks to form a better soft prompt initialization, and thus adapts

faster and better across various target domain tasks (See Figure 1). Extensive experiments show that MetaPrompting achieves promising performance with desired robustness.

We summarize the main contribution of this paper as follows:

(1) We propose a novel prompting method MetaPrompting, which employs optimization-based meta-learning algorithm to find adaptive initialization for soft-prompt methods. To the best of our knowledge, this is the first study of applying meta-learning to prompting problem setting.

(2) We conduct extensive experiments on four different datasets with various few-shot settings, and results show the superiority of MetaPrompting over normally fine-tuned soft-prompt methods and SOTA meta-learning baselines.

(3) Further analysis experiments indicate that MetaPrompting significantly alleviates soft prompt initialization problem, and learns general meta-knowledge to counter the instability of prompt variance. We also study MetaPrompting's compatibility with different meta-learning methods and give empirical analysis of their performance difference.

All code and data will be publicly available at https://github.com/Dousia/MetaPrompting.

2 Preliminaries and Related Works

In this section, we review related work and provide preliminaries about Language Model Prompting and Meta-learning.

2.1 Prompting Language Models

Prompting methods are proposed to better apply pre-trained language models to downstream tasks by aligning them with pre-training tasks. For Masked Language Models (MLMs), the first step is to convert a sample text x to x_{prompt} by inserting prompt words which contain [MASK] tokens. Taking the news headline classification task as an example, the prompted text is given as:

$$x_{prompt} = \texttt{[CLS]} \; x \; \text{The topic is [MASK] . [SEP]}, \quad (1)$$

where "The topic is [MASK]" are prompt tokens. Then, we ask pre-trained MLM to complete the prompted text x_{prompt}, and the word to be filled at [MASK] position is regarded as an answer. An answer-label map is then used to convert the word probability distribution at [MASK] to classification results. For example, answers 'arts' and 'culture' can be mapped to label 'ARTS & CULTURE',

while 'environment' can be mapped to label 'EN-VIRONMENT'. The average probability of each label's corresponding answers is computed as the label's final probability.

Early prompting methods, such as GPT-3 (Brown et al., 2020) and PET/iPET (Schick and Schütze, 2021), use hand-crafted prompt templates. Although promising results are achieved, the performance of these methods heavily relies on the selection of pre-defined prompt templates. Moreover, designing prompts is extremely time-consuming, and hand-crafted prompts may be sub-optimal.

A number of recent works propose to automate the search of discrete prompt templates (Shin et al., 2020; Gao et al., 2021; Davison et al., 2019; Jiang et al., 2020; Haviv et al., 2021), while others treat prompt tokens as continuous trainable parameters (Li and Liang, 2021; Liu et al., 2021c; Qin and Eisner, 2021). In this work, we follow P-tuning (Liu et al., 2021c) to combine soft prompt and anchor tokens as templates. Instead of directly applying the model in few-shot tasks, however, we adopt meta-learning methods to find a better initialization point for both soft prompt embeddings and MLM parameters, because they are very sensitive to initialization in few-shot settings (Li and Liang, 2021; Liu et al., 2021b). Note that a recent work (Zhong et al., 2021a) also learns prompt model on a number of source domain tasks, but their method consumes heavy human labor to design hard prompts for each task, and directly fine-tunes the model without involving meta algorithms.

2.2 Meta Learning

Meta-learning algorithms can be classified into metric-based methods, model-based methods and optimization-based methods. Metric-based methods such as Siamese Network (Koch et al., 2015), Matching Network (Vinyals et al., 2016) and Prototypical Network (Snell et al., 2017), are proposed to learn a metric space that gathers similar samples and separates distinct ones. Model-based meta-learning algorithms use additional meta learners to assist model prediction (Graves et al., 2014; Mishra et al., 2018; Qiao et al., 2018).

Different from above algorithms, optimization-based meta-learning methods learn to improve model's optimization procedure. Optimization-based approach is more suitable for prompting models as it neither requires a specific task form (i.e., metric learning form) nor additional archi-
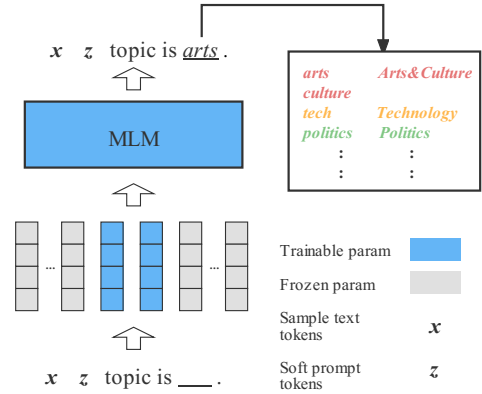

Figure 2: An illustration of soft-prompting method. x refers to the sample text, and z represents soft prompt tokens. All trainable parameters are colored in blue, while fixed ones are colored in grey.

tecture (e.g. memory-augmented components in model-based algorithms). Andrychowicz et al. (2016) and Ravi and Larochelle (2017) train recurrent neural networks to transform vanilla gradient descent direction for better optimization results. MAML (Finn et al., 2017) optimizes model parameters to find a better initialization point, so that the model can adapt faster and better to unseen tasks. In addition to MAML, more elaborate methods also learn inner loop gradient descent direction (Li et al., 2017) and inner step sizes (Antoniou et al., 2019). Utilizing first-order derivatives, FOMAML (Finn et al., 2017) and Reptile (Nichol et al., 2018) are proposed to reduce the memory consumption of high-order derivative calculation.

3 Method

Since prompt-based methods, especially those adopting soft prompts, are very sensitive to parameter initialization (Li and Liang, 2021; Liu et al., 2021b), we introduce optimization-based meta-learning methods into prompting methods to find better initialization points for prompt-based models and further explore their capabilities in few-shot scenarios. In this section, we first introduce the prompt-based model tuning process used in our method (§3.1), and then describe how to construct Meta Prompting tasks (§3.2). Finally, we elaborate and formulate the Meta Prompting objective and parameter updating strategies (§3.3 and §3.4).

3.1 Prompt-based Model Tuning

In this work, we use soft prompts with anchor tokens. As illustrated in Figure 2, prompt tokens consist of soft tokens which are represented as

trainable parameters (blue) as well as anchor tokens which are fixed as the embeddings of specific words (grey). Hard-soft combined prompt templates render the model more flexible, while preserving enough semantic information to trigger the MLM to produce correct predictions. Similar to P-tuning (Liu et al., 2021c), we implement transformation layers on soft prompt embeddings, allowing them to escape from local minima smoothly.

In this way, we define MLM parameters as θ and soft prompt token embeddings as ϕ. Given a few-shot task τ where $\mathcal{D}_\tau = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i \in \tau}$ represents training samples, the prompt tuning objective can be formulated as follows:

$$
\begin{aligned}
\theta^*, \phi^* &= \underset{\theta,\phi}{\arg\min}\, \mathcal{L}_{\mathcal{D}_\tau}(f_{\phi,\theta}) \\
&= \underset{\theta,\phi}{\arg\max} \sum_{(\mathbf{x}_i,\mathbf{y}_i) \in \mathcal{D}_\tau} \log P(\mathbf{y}_i|\mathbf{x}_i; \phi, \theta),
\end{aligned} \quad (2)
$$

where $\mathcal{L}$ is the loss function, and $f_{\phi,\theta}$ is prompt-based model parameterized by MLM parameters θ and soft prompt embeddings ϕ.

$\mathcal{D}_\tau$ contains few labeled data because of the high annotation cost in real-world scenarios. As a result, the initialization of parameters θ and ϕ are more than crucial to the model's performance.

3.2 Constructing Meta Prompting Tasks

To get a better initialization point for parameters θ and ϕ, we propose to sample Meta Prompting tasks from accessible source data and conduct meta-training on these sampled tasks. This meta training process aims to simulate the model's adaptation to new few-shot tasks.

We sample each Meta Prompting task τ_i as:

$$
\tau_i = (\mathcal{D}_{\tau_i}^{support}, \mathcal{D}_{\tau_i}^{query}), \quad (3)
$$

where $\mathcal{D}_{\tau_i}^{support}$ indicates the support set and $\mathcal{D}_{\tau_i}^{query}$ indicates the query set in traditional few-shot learning settings. Note that meta training tasks and meta testing tasks should be sampled from different domains, to prevent the model from simply memorizing training samples.

3.3 Applying Meta-learning to Prompting Models

After constructing Meta Prompting tasks, we train our prompting model on these tasks to find a better initialization point. Figure 3 illustrates the meta training and meta testing procedures of MetaPrompting. Given a Meta Prompting task τ_i, we clone the model's parameters and simulate the
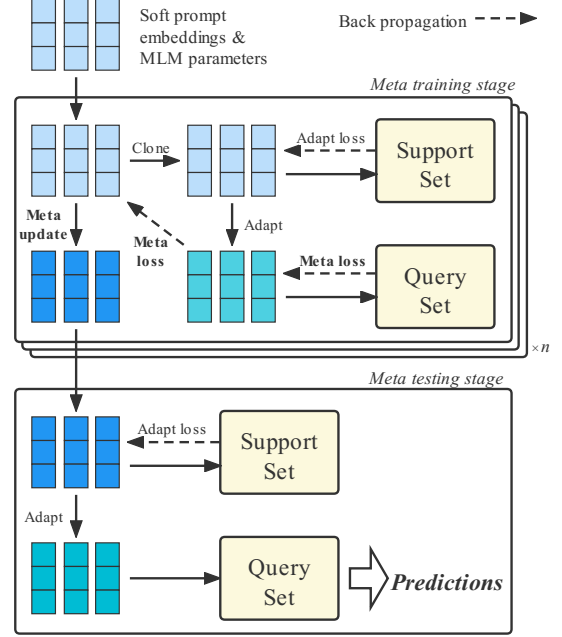

Figure 3: Meta training and testing procedures of MetaPrompting.

adaption process of few-shot tasks by updating cloned model parameters θ_i^0 and ϕ_i^0 on $\mathcal{D}_{\tau_i}^{support}$. The adaption objective is given in Equation (2), and this process can be formulated as:

$$
\begin{aligned}
\theta_i^k &= \theta_i^{k-1} - \alpha \nabla_{\theta_i^{k-1}} \mathcal{L}_{\mathcal{D}_{\tau_i}^{support}}(f_{\phi_i^{k-1}, \theta_i^{k-1}}), \\
\phi_i^k &= \phi_i^{k-1} - \alpha \nabla_{\phi_i^{k-1}} \mathcal{L}_{\mathcal{D}_{\tau_i}^{support}}(f_{\phi_i^{k-1}, \theta_i^{k-1}}),
\end{aligned} \quad (4)
$$

where α indicates learning rate and $k = 1, 2, 3, \ldots$ indicates the inner step. The goal of learning with Meta Prompting tasks is to minimize the loss of the adapted prompting model, which is parameterized as f_{ϕ_i, θ_i}, on $\mathcal{D}_{\tau_i}^{query}$. This objective can be described as follows:

$$
\theta^{obj}, \phi^{obj} = \underset{\theta_i, \phi_i}{\arg\min}\, \mathcal{L}_{\mathcal{D}_{\tau_i}^{query}}(f_{\phi_i, \theta_i}). \quad (5)
$$

Optimizing towards this objective is to mimic real few-shot text classification scenarios, and enable prompting model to find a better initialization point for fast adaptation to new tasks. Let β be the learning rate when updating model parameters on $\mathcal{D}_{\tau_i}^{query}$, and $\mathbf{H}$ be Hessian matrix. We formulate the second-order gradient of prompt parameter ϕ computed on $\mathcal{D}_{\tau_i}^{query}$ in the following form:

$$
\begin{aligned}
\phi &\leftarrow \phi - \beta \cdot g_\phi^{second} \\
&= \phi - \beta \nabla_\phi \mathcal{L}_{\mathcal{D}_{\tau_i}^{query}}(f_{\phi_i, \theta_i}) \\
&= \phi - \beta \nabla_{\phi_i} \mathcal{L}_{\mathcal{D}_{\tau_i}^{query}}(f_{\phi_i, \theta_i}) \cdot \nabla_\phi(\phi_i) \\
&= \phi - \beta \nabla_{\phi_i} \mathcal{L}_{\mathcal{D}_{\tau_i}^{query}}(f_{\phi_i, \theta_i}) \cdot \\
&\quad (\mathbf{I} - \alpha \mathbf{H}_\phi(\mathcal{L}_{\mathcal{D}_{\tau_i}^{support}}(f_{\phi, \theta}))),
\end{aligned} \quad (6)
$$

where we assume ϕ_i is ϕ adapted for one inner step on $\mathcal{D}_{\tau_i}^{support}$. In practice, inner steps can be increased for better performance. Pre-trained MLM parameters θ is updated in the same way as prompt parameters ϕ in Equation (6).

3.4 Stable and Memory-efficient Meta Prompt Learning

Although broadly used in meta-learning tasks, MAML suffers from training instability and exploding memory consumption when model size and inner step grow. To address the first problem, we follow Antoniou et al. (2019) to introduce Multi-Step Loss Back-propagation (MSLB) into prompting model tuning process. In this way, prompting model parameters receive optimization information from each inner step during adaptation, alleviating the vanishing/exploding gradient problem in the stacked deep neural architecture constructed in adaptation process.

As for the exploding memory consumption issue, we also explore to combine memory-efficient alternatives such as FOMAML (Finn et al., 2017) and Reptile (Nichol et al., 2018) with prompting model. FOMAML removes the high-order derivatives term in Equation (6), providing a cheap approximation for MAML. Reptile updates model parameters towards the optimal point of each task, which is obtained by adapting the model on the support set samples. Equipped with these algorithms, MetaPrompting can learn meta knowledge with limited memory consumption.

4 Experiment

We conduct experiments by evaluating the proposed methods on four widely-used benchmark datasets with various low resource settings.

4.1 Dataset

Following Bao et al. (2019); Xu and Xiang (2021), we use the following four text classification datasets for experiments, which provide well-founded benchmarks for the meta-train & meta-test setting and vary in domain and text length.

HuffPost headlines contains around $36,900$ news headlines from 2012 to 2018 obtained from HuffPost (Misra, 2018; Misra and Grover, 2021).

These headlines cover 41 news categories and the average text length is 11.

Amazon product data contains around $24,000$ product reviews from 1996 to 2014 from Amazon (He and McAuley, 2016). These reviews contain 24 categories corresponding to their respective product categories with varying text lengths. The average text length is 140.

20 Newsgroups contains $18,820$ newsgroup documents of 20 different topics (Lang, 1995). We used 20news-18828 version following Bao et al. (2019). The average text length is 340.

Reuters contains 620 document-level news articles of 31 different topics from 1987 (Lewis, 1997). The average text length is 168.

We adopt the same pre-processing and data-splitting strategy with Bao et al. (2019) to process the above datasets.

4.2 Implementation

We use the pre-trained BERT (bert-base-uncased) with HuggingFaces codebase (Wolf et al., 2019) as the pre-trained language model.

For soft prompting model, we follow Liu et al. (2021c) to use a two-layer biLSTM and a two-layer MLP to transform soft-prompt embeddings. We divide the learnable parameters of prompting model into two parts: pre-trained model and prompt embeddings. AdamW (Loshchilov and Hutter, 2018) is used to optimize two types of parameters, with initial learning rates of $1e-5$ and $5e-5$, respectively. For pre-trained model parameters, we set weight decay to 0.1. We also adopt linear warmup and linear decay strategy for learning rates. Batch size is set as 16 for all stages, and the model adapts for 15 epochs on test episodes. We run 3 independent runs with random seeds for each setting.

Before meta-training stage, we generate $10,000$ training episodes, $2,500$ validation episodes and $1,000$ testing episodes comprehensively and randomly. During the training stage, we train the model with 100 sampled training episodes per epoch. When there is no validation accuracy increase for 10 epochs, we apply early stopping. For meta-testing, we test the model on all $1,000$ test episodes and report the average accuracy.

4.3 Baselines

We compare with the following baselines:

1-NN is a 1-nearest-neighbor classifier based on Euclidean distance.

For other datasets used in Bao et al. (2019), RCV1 (Lewis et al., 2004) is not included due to overly long text lengths, while FewRel (Han et al., 2018) is excluded because it does not provide each label's semantic meanings.

Method	20 News		Amazon		HuffPost		Reuters		Average	
	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot
1-NN	38.8	51.9	51.4	67.1	31.5	42.3	57.8	82.9	44.88	61.05
FT	33.0	47.1	45.7	63.9	32.4	44.1	70.9	91.0	45.50	61.53
PROTO	37.8	46.5	41.9	59.2	34.8	50.2	61.0	72.1	43.88	57.00
MAML	37.2	48.6	43.6	62.4	38.9	53.7	61.5	72.0	45.30	59.18
RR	44.8	64.3	60.2	79.7	37.6	59.5	69.1	93.0	52.93	74.13
DS (2019)	52.1	68.3	62.6	81.1	43.0	63.5	81.8	96.0	59.88	77.23
DE-MLMAN (2021)	–	–	–	–	49.7	60.9	–	–	–	–
DE-MAML (2021)	–	–	–	–	51.8	67.3	–	–	–	–
DE-PROTO (2021)	–	–	–	–	52.3	69.6	–	–	–	–
KGML-PROTO (2021)	–	–	58.6	74.5	42.3	58.7	–	–	–	–
KGML-MAML (2021)	–	–	51.4	58.8	44.2	54.1	–	–	–	–
P-TUNING (2021c)	56.21	77.70	62.18	79.13	54.48	65.75	90.01	96.66	65.72	79.81
FROG-GNN (2021)	–	–	71.5	83.6	54.1	69.6	–	–	–	–
LaSAML-PN (2021)	–	–	–	–	62.1	70.1	–	–	–	–
CONTRASTNET (2022)	**71.74**	81.57	76.13	85.17	53.06	65.32	86.42	95.33	71.84	81.85
OURS (PRETRAIN INIT)	68.42	79.11	75.12	83.27	70.82	75.47	95.07	**97.29**	77.36	83.79
OURS (META INIT)	68.83	**82.95**	**77.65**	**85.54**	**71.93**	**76.32**	**95.20**	97.17	**78.40**	**85.50**

Table 1: Results of 1-shot and 5-shot classification on four datasets in terms of accuracy. The rows below the mid-line are results of MetaPrompting. '-' means that the result of this dataset is not given in the original paper. We do not show standard deviation of our experiment results here due to space limits. Full results can be found in Appendix A.

FT (Chen et al., 2019) pre-trains a classifier on source domain data, and then fine-tunes (FT) it on each support set before evaluation.

RR (Bertinetto et al., 2019) adopts ridge regression (RR) for classification.

MAML (Finn et al., 2017) meta-learns a classifier with MAML algorithm, so that the model can adapt faster and better to target domain tasks.

Prototypical network (Snell et al., 2017) is a metric-based method which meta-learns a metric space by minimizing the Euclidean distance between the centroid of each category and the corresponding samples.

DS (Bao et al., 2019) is trained within a meta-learning framework to map the distribution signatures (DS), i.e., characteristics of the underlying word distributions, into attention scores to extract more transferable features.

DE (Ohashi et al., 2021) generates distinct label representations that embed information specific to each label to aid classification tasks. During experiments, it is combined with MAML (DE-MAML) and prototypical network (DE-PROTO), as well as MLMAN (Ye and Ling, 2019) (DE-MLMAN).

KGML (Yao et al., 2021) extracts additional representation for each sentence from external knowledge base, to bridge the gap between meta-training and meta-testing tasks. During experiments, it

---

The above 6 baselines uses fastText embeddings (Joulin et al., 2016) and each word's inverse document frequency to produce sentence embeddings.

---

works with MAML (KGML-MAML) and prototypical network (KGML-Proto).

P-tuning (Liu et al., 2021c) is a prompt-based method that uses masked language model to convert target tasks into cloze problems. It employs soft-prompting techniques to optimize prompts in continuous space.

Frog-GNN (Xu and Xiang, 2021) is a graph neural network based method, which extracts better query representations with multi-perspective aggregation of graph node neighbors.

LaSAML-PN (Luo et al., 2021) is a meta-learning framework that mines semantic information in labels and attaches it to the sentence as the input of the encoder to obtain discriminative sentence embeddings.

ContrastNet (Chen et al., 2022) is the SOTA method. It introduces instance-level and task-level regularization loss into vanilla contrastive learning model based on BERT representations for better generalization performance. The regularization loss is computed with samples augmented by an additional BERT model.

4.4 Main Results

We evaluate the proposed methods in both 5-way 1-shot and 5-way 5-shot settings and report performance on four different datasets with different text styles. As shown in Table 1, our model outperforms previous SOTA method ContrastNet without using additional PLM. Averagely, our model

improves 1-shot accuracy by 6.56 (9.13% ↑) and 5-shot accuracy by 3.65 (4.46% ↑) across four datasets. MetaPrompting gains less improvement on 20Newsgroup and Amazon, because their labels are hard to interpret as natural words, which poses difficulties for prompting models (Shin et al., 2020; Cui et al., 2022). Various methods are proposed to address this problem (Shin et al., 2020; Gao et al., 2021; Hambardzumyan et al., 2021; Cui et al., 2022; Jiang et al., 2021). In this work, however, we do not spend much effort on elaborate answer design but instead focus on soft prompt initialization problem. Although equipped with simply designed answer sets, MetaPrompting still achieves new state-of-the-art performance across the four datasets.

Meanwhile, we have following observations based on Table 1:

(1) Compared with other soft-prompting methods, i.e., P-tuning, our method obtains superior results. Although meta learning is conducted on completely different source domain Meta Prompting tasks, our method still learns a better prompt model initialization point, which allows faster and better adaption to new prompting tasks.

(2) When compared to traditional supervised learning methods, such as FT, all prompt-based methods achieve significant improvements, which demonstrates the effectiveness of prompting mechanism in narrowing the gap between pretraining and downstream tasks.

(3) Metric learning-based baselines, such as ContrastNet and LASAML-PN, perform as the strongest baselines on Amazon and HuffPost datasets, respectively. We find that directly using prompt-based method may not necessarily perform better, because of the absence of domain-related initialization. The proposed MetaPrompting alleviates the above issue and achieves new state-of-the-art performance. Among strong metric-learning baselines, Frog-GNN conducts transductive learning with additional label propagation information, and ContrastNet uses an additional BERT model to regularize the main model with augmented data. Our model achieves better performance without implementing any of above tricks.

(4) Compared with other optimization-based meta-learning methods such as MAML, DE-MAML and KGML-MAML, MetaPrompting consistently performs better, demonstrating good compatibility between prompting methods and meta-

Method	HuffPost	Amazon	20 News	Reuters
BASELINE	65.75	79.13	77.70	96.66
OURS	73.06	83.64	81.79	97.26

Table 2: PLM frozen, MetaPrompting still achieves better performance over randomly initialized baseline on test domains. Results are given in 5 shot setting.

learning. Note that KGML-MAML adopts an additional knowledge base, while our model does not but still achieves better performance.

(5) To further demonstrate meta learning method's effectiveness in prompt learning, we conduct ablations study by removing the Meta Prompting Objective and learn an initialization by pretraining soft prompt model on Meta Prompting Tasks described in Section 3.2. The results are shown as OURS (PRETRAIN INIT). Performance drops are witnessed across all four datasets and few-shot settings, validating the necessity of meta objectives in finding a better initialization.

4.5 Analysis

In this part, we analyze the proposed method from different aspects.

MetaPrompting tackles soft prompt initialization problem. Main results displayed in Section 4.4 are obtained with LM parameters tuned for fair comparison with previous SOTA baselines. To further validate the importance of learning a good prompt initialization, we freeze PLM's parameters while leaving soft prompt parameters unfrozen to only learn a better prompt initialization on source domains. We test our meta-learning-based initialization strategy against random initialization, and the results are shown in Table 2. The randomly initialized soft prompt baseline performs poorly and unstably, while our method consistently yields better results with lower variance across four datasets, which verifies our hypothesis and the validity of the MetaPrompting.

MetaPrompting learns general meta-knowledge from various source domains. Although MetaPrompting shows promising results when meta trained on source domain tasks from the same dataset, it is impractical to always build corresponding meta-training tasks for each few-shot scenario. To this end, we conduct meta-training on Out-Of-Domain (OOD) tasks to better analyze MetaPrompting's ability in

Source domain	Target domain	Acc
–	HuffPost	65.75
Metatuning	HuffPost	67.46
Reuters	HuffPost	69.18
20 Newsgroup	HuffPost	71.04
Amazon	HuffPost	71.47
HuffPost (Diff. label set)	HuffPost	76.32

Table 3: Given irrelevant source domain data, MetaPrompting still learns meta knowledge to improve the performance on target domains.

transferring meta-knowledge from various source domains.

Table 3 shows the results of 5-shots setting. Even given irrelevant meta-training data and prompt templates from other datasets, MetaPrompting still learns meta knowledge to tackle target domain tasks and outperforms the baseline robustly. Among OOD datasets, Metatuning (Zhong et al., 2021a) contains a series of text classification tasks, and each task is accompanied by several hand-crafted questions which require yes/no answers. The task formulation of Metatuning is distinct from HuffPost. However, MetaPrompting still makes it to transfer meta-knowledge from Metatuning to HuffPost's target domains, improving model performance by approximately 2 points. Although MetaPrompting's performance varies among source domain tasks according to their data quality for generalization purposes, the proposed model outperforms the baseline across all source domain tasks, verifying MetaPrompting's effectiveness in transferring meta-knowledge.

Anti-disturbance analysis We expect the meta-learned initialization alleviates prompting models' susceptibility to varying prompt forms. To verify this, we test the prompting model with multiple different prompt forms and report the standard deviation. Specifically, we add two more discrete prompt templates, and randomly replace the template tokens with pseudo tokens to test MetaPrompting's robustness across different templates.

Table 4 shows the results. While changing the prompting form indeed impacts the performance for both our method and normal soft prompting methods, the proposed meta-learning method significantly reduces performance fluctuation, showing impressive anti-disturbance ability. Therefore,

| Method | HuffPost | | Amazon | |
	1 shot	5 shot	1 shot	5 shot
P-TUNING	±3.46	±1.90	±5.30	±1.85
OURS	±0.23	±0.09	±0.17	±0.45

Table 4: Analysis for anti-disturbance against changing of prompting form.

Setting	MAML++	MAML	FOMAML	Reptile
1 SHOT	71.93	71.43	70.56	69.76
5 SHOT	76.32	76.04	76.08	74.09

Table 5: MetaPrompting's performance with different meta learning methods.

the proposed method is promising in real-world applications, because prompt designing requires heavy workload and domain-specific knowledge. Applying MetaPrompting significantly reduces the cost of prompt engineering.

Applying different meta-learning methods to prompting models. In this part, we conduct empirical analysis on different optimization-based meta-learning methods applied in prompting models. Results are shown in Table 5. Stabilizing MAML training procedure, MAML++ performs the best among all methods, while Reptile fails to achieve comparable performance with others. We attribute Reptile's low performance to PLM's sensitivity to parameter tuning process, which can be distorted by Reptile's parameter updating strategy. MAML and FOMAML show similar results, because MetaPrompting's slow tuning process narrows the gap between their calculated gradient during meta-training.

Analysis for learning procedure of prompting methods. We analyze the decreasing trend of adaptation loss to better understand the learning procedure of soft-prompt model. Specifically, we visualize model adaptation loss curve during meta-testing on 5 shot Amazon dataset.

As shown in Figure 4, task-related initialization (Ours (Pretrain Init)) helps the model converge faster and end up at a lower position than randomly initialized baseline. The proposed meta-learning-based method (Ours (Meta Init)) further improves the learning process in new tasks, indicating that the meta-learned initialization point contains more

---

We add "*The topic/product category: [MASK]. Input: x*" and "*x. What is the topic/product category ? [MASK].*", where topic and product category are used for HuffPost and Amazon dataset respectively.

We only include the MSLB trick of MAML++ (Antoniou et al., 2019) due to the incompatibility (BN layer tricks) or trivial performance improvement (Per-step adaption loss, cosine annealing learning rates).
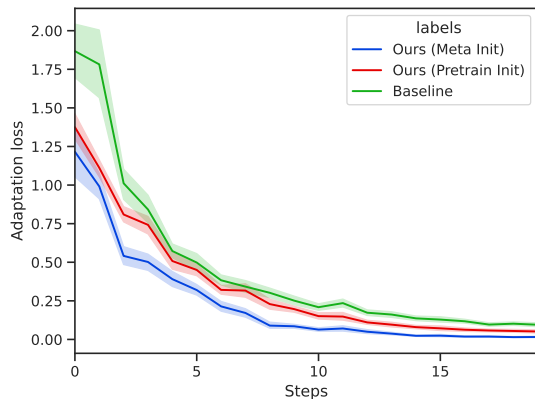
Figure 4: Analysis on prompt learning process.

generalizable meta knowledge to aid new tasks.

5 Conclusion

In this paper, we introduce a generalized optimization-based meta-learning approach MetaPrompting for few-shot NLP problems. Utilizing sampled meta tasks and meta-learning-based optimization, MetaPrompting learns to find an initialization that alleviates soft prompt initialization problem, and allows better and faster adaption to new tasks. Extensive experiments on four few-shot learning benchmarks show that MetaPrompting significantly outperforms vanilla soft-prompting models and strong meta-learning baselines, achieving new state-of-the-art performance.

6 Acknowledgement

References

Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *Proc. of NIPS*.

Antreas Antoniou, Harri Edwards, and Amos Storkey. 2019. How to train your maml. In *Proc. of ICLR*.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2019. Few-shot text classification with distributional signatures. In *Proc. of ICLR*.

L Bertinetto, J Henriques, PHS Torr, and A Vedaldi. 2019. Meta-learning with differentiable closed-form solvers. In *Proc. of ICLR*.

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. In *Proc. of NIPS*, volume 33, pages 1877–1901.

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xue. 2022. Contrastnet: A contrastive learning framework for few-shot text classification. In *Proc. of AAAI*.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. In *Proc. of ICLR*.

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7014–7024.

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proc. of EMNLP-IJCNLP*, pages 1173–1178.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. of ICML*, pages 1126–1135.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proc. of ACL-IJCNLP*, pages 3816–3830.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. Warp: Word-level adversarial reprogramming. 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 4921–4933.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proc. of EMNLP*, pages 4803–4809.

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. BERTese: Learning to speak to BERT. In *Proc. of EACL*, pages 3618–3623.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proc. of WWW*, pages 507–517.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proc. of ICML*, pages 331–339.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proc. of EMNLP*, pages 3045–3059.

D Lewis. 1997. Reuters-21578 text categorization test collection. *Distribution 1.0, AT&T Labs-Research*.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5:361–397.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proc. of ACL-IJCNLP*, pages 4582–4597.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021a. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *Proc. of ICLR*.

Qiaoyang Luo, Lingqiao Liu, Yuhao Lin, and Wei Zhang. 2021. Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification. In *Findings of ACL-IJCNLP*, pages 2773–2782.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2018. A simple neural attentive meta-learner. In *Proc. of ICLR*.

Rishabh Misra. 2018. News category dataset, 06.

Rishabh Misra and Jigyasa Grover. 2021. *Sculpting Data for ML: The first act of Machine Learning*.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Sora Ohashi, Junya Takayama, Tomoyuki Kajiwara, and Yuki Arase. 2021. Distinct label representations for few-shot text classification. In *Proc. of ACL-IJCNLP*, pages 831–836.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. 2018. Few-shot image recognition by predicting parameters from activations. In *Proc. of CVPR*, pages 7229–7238.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proc. of NAACL-HLT*, pages 5203–5212.

Sachin Ravi and H. Larochelle. 2017. Optimization as a model for few-shot learning. In *Proc. of ICLR*.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proc. of COLING*, pages 5569–5578.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proc. of EACL*, pages 255–269.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proc. of NAACL*, pages 2339–2352.

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 *Proc. of EMNLP*, pages 4222–4235.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proc. of NIPS*, volume 30, pages 4077–4087.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Proc. of NIPS*, volume 29, pages 3630–3638.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Shiyao Xu and Yang Xiang. 2021. Frog-gnn: Multi-perspective aggregation based graph neural network for few-shot text classification. *Expert Systems with Applications*, 176:114795.

Huaxiu Yao, Ying-xin Wu, Maruan Al-Shedivat, and Eric Xing. 2021. Knowledge-aware meta-learning for low-resource text classification. In *Proc. of EMNLP*, pages 1814–1821.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Multi-level matching and aggregation network for few-shot relation classification. In *Proc. of ACL*, pages 2872–2881.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021a. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of EMNLP*, pages 2856–2878.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021b. Factual probing is [MASK]: learning vs. learning to recall. In *Proc. of NAACL-HLT*, pages 5017–5033.

A   Full experiment results

Here we provide full experiment results with standard deviation. Full results are shown in Table 6.

Method	20 News		Amazon		HuffPost		Reuters		Average	
	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot
1-NN	38.8	51.9	51.4	67.1	31.5	42.3	57.8	82.9	44.88	61.05
FT	33.0	47.1	45.7	63.9	32.4	44.1	70.9	91.0	45.50	61.53
PROTO	37.8	46.5	41.9	59.2	34.8	50.2	61.0	72.1	43.88	57.00
MAML	37.2	48.6	43.6	62.4	38.9	53.7	61.5	72.0	45.30	59.18
RR	44.8	64.3	60.2	79.7	37.6	59.5	69.1	93.0	52.93	74.13
DS (2019)	52.1	68.3	62.6	81.1	43.0	63.5	81.8	96.0	59.88	77.23
DE-MLMAN (2021)	—	—	—	—	49.7	60.9	—	—	—	—
DE-MAML (2021)	—	—	—	—	51.8	67.3	—	—	—	—
DE-PROTO (2021)	—	—	—	—	52.3	69.6	—	—	—	—
KGML-PROTO (2021)	—	—	58.6	74.5	42.3	58.7	—	—	—	—
KGML-MAML (2021)	—	—	51.4	58.8	44.2	54.1	—	—	—	—
P-TUNING (2021c)	56.21	77.70	62.18	79.13	54.48	65.75	90.01	96.66	65.72	79.81
FROG-GNN (2021)	—	—	71.5	83.6	54.1	69.6	—	—	—	—
LASAML-PN (2021)	—	—	—	—	62.1	70.1	—	—	—	—
CONTRASTNET (2022)	**71.74**	81.57	76.13	85.17	53.06	65.32	86.42	95.33	71.84	81.85
OURS (PRETRAIN INIT)	68.42±0.06	79.11±0.79	75.12±0.65	83.27±0.03	70.82±0.74	75.47±0.24	95.07±0.15	**97.29**±0.06	77.36	83.79
OURS (META INIT)	68.83±0.21	**82.95**±0.12	**77.65**±0.47	**85.54**±0.03	**71.93**±0.22	**76.32**±0.05	**95.20**±0.12	97.17±0.18	**78.40**	**85.50**

Table 6: Full results of 1-shot and 5-shot classification on four datasets in terms of accuracy. The rows below the mid-line are results of MetaPrompting. '-' means that the result of this dataset is not given in the original paper.