Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

TransPrompt v2: Transferable Prompt-based Fine-tuning for Few-shot Text Classification

Jianing Wang (lygwjn@gmail.com)
East China Normal University
Chengyu Wang
Alibaba Group (China)
Cen Chen
East China Normal University
Ming Gao
East China Normal University
Jun Huang
Alibaba Group (China)
Aoying Zhou
East China Normal University

Research Article

Keywords: Few-shot learning, Text classification, Prompt-based fine-tuning, Transfer learning, Meta learning

Posted Date: June 9th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1711263/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

TransPrompt v2: Transferable Prompt-based Fine-tuning for Few-shot Text Classification

Jianing Wang[1†], Chengyu Wang[2†], Cen Chen[1], Ming Gao[1*], Jun Huang[2] and Aoying Zhou[1]

[1*]School of Data Science and Engineering, East China Normal University, Shanghai, 200062, China.
[2]Alibaba Group, Hangzhou, 311121, China.

*Corresponding author(s). E-mail(s): mgao@dase.ecnu.edu.cn;
Contributing authors: lygwjn@gmail.com;
chengyu.wcy@alibaba-inc.com; cenchen@dase.ecnu.edu.cn;
huangjun.hj@alibaba-inc.com; ayzhou@dase.ecnu.edu.cn;
†These authors contributed equally to this work.

Abstract

Recent studies have shown that prompt-based fine-tuning improves the performance of large Pre-trained Language Models (PLMs) for few-shot text classification. Specifically, this type of method transforms the text classification task into the inherent Masked Language Modeling (MLM) with some natural language prompts and verbalizers to bridge the gap between the pre-training stage and the fine-tuning stage. Yet, it is unclear how the prompting knowledge can be transferred across similar or distant NLP tasks, for the purpose of mutual reinforcement. Based on continuous prompt embeddings, we propose *TransPrompt v2*, a novel transferable prompting framework for few-shot learning across similar or distant text classification tasks. For transferable prompt learning across similar tasks, we employ a multi-task meta-knowledge acquisition procedure to train a meta-learner that captures the cross-task transferable knowledge. For learning across distant tasks, we further introduce the task type descriptions and propose intra-type and inter-type prompt encoders to capture implicit mutual relations among multiple distant tasks. Additionally, two de-biasing techniques are further designed to make the trained meta-learner more task-agnostic and unbiased towards

any tasks. After that, the meta-learner can be fine-tuned for specific tasks with better parameters initialization. Extensive experiments show that *TransPrompt v2* outperforms single-task and cross-task strong baselines over multiple NLP tasks and datasets. We further show that the meta-learner can effectively improve the performance of PLMs on previously unseen tasks. In addition, *TransPrompt v2* also outperforms strong fine-tuning baselines when learning with full training sets. *

Keywords: Few-shot learning, Text classification, Prompt-based fine-tuning, Transfer learning, Meta learning.

1 Introduction

Pre-trained Language Models (PLMs) have become the dominant infrastructures for a majority of NLP tasks by employing the standard two-stage training, i.e., *pre-training* and *fine-tuning*. Notable PLMs include BERT [2], RoBERTa [3], ALBERT [4], XLNet [5], GPT-2 [6] and many others. In the *pre-training* stage, these PLMs follow the Masked Language Modeling (MLM) objective or Autoregressive Language Modeling (ALM) over large-scale unlabeled corpora to capture the intrinsic semantics of the natural language. While in the *fine-tuning* stage, task-specific prediction heads are additionally stacked over PLMs to train on downstream tasks. Unfortunately, several studies reveal that the vanilla PLM fine-tuning approach over very few data samples does not always guarantee satisfactory results [7, 8]. Due to the existence of many newly initialized parameters in the prediction head, PLMs can fall into the over-fitting problem in low-resource scenarios.

Recently, the mature ultra-large PLM model GPT-3 [9] has been proposed to enable in-context learning with some discrete templates in low-resource settings. Inspired by these works, Gao et al. [7] propose a prompt-based approach to fine-tune BERT-style PLMs in a few-shot learning setting, which adapts PLMs into producing specific tokens corresponding to each class, instead of learning the prediction head. The effectiveness of prompts has also been shown in [10–12] and others. However, designing high-performing prompts is challenging and requires a very large validation set. To alleviate this problem, Liu et al. [8] propose *continuous prompt embeddings* with fully differentiable parameters, avoiding the cumbersome manual prompt engineering process.

Despite the remarkable success, we notice that current prompt-based approaches may have a few limitations. For few-shot learning, the performance of downstream tasks is still constrained by the number of training instances from a single task. It would be highly desirable if the model can acquire the transferable knowledge from other NLP tasks before it is adapted to specific tasks with few samples. For PLMs, Wang et al. [13] discover that training a

*An earlier version [1] of this paper was presented at the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021).

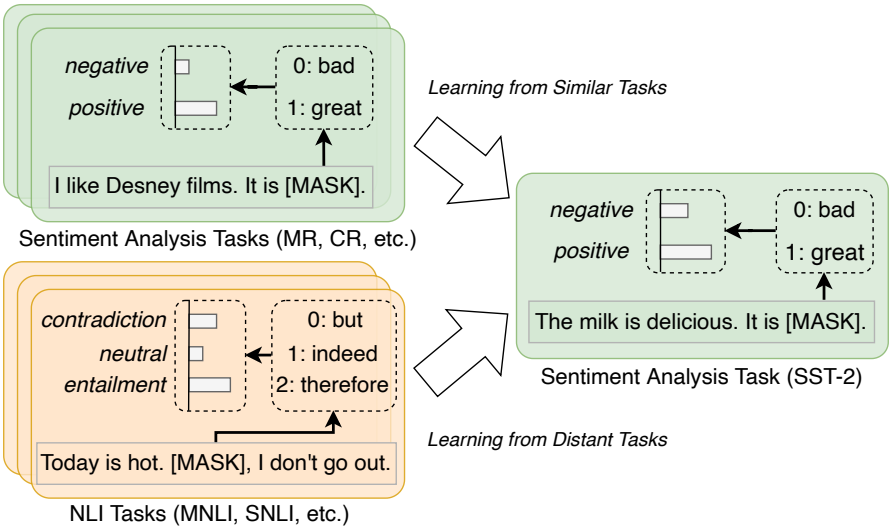


Fig. 1 Example of prompt-based learning across similar and distant tasks. The sentiment analysis task SST-2 can benefit from similar tasks such as MR and CR, and also from distant tasks such as MNLI and SNLI.

meta-learner for PLMs is effective to capture the transferable knowledge across domains. Wang et al. [14] consider the transferable of PLM over multiple distant tasks. Yet, these methods are not designed for prompt-based learning and lack the mechanism to learn *unbiased representations* for all the tasks. Therefore, a natural question arises: *how can we design a prompting framework for BERT-style models that captures transferable knowledge across both similar and distant NLP tasks to improve the performance of few-shot learning*?

In this paper, we extend our previous work [1] and present *TransPrompt v2*, a general prompting framework that allows PLMs to capture *cross-task transferable knowledge* for few-shot text classification. Specifically, to improve the application scope of our algorithm, *TransPrompt v2* considers two scenarios shown in Figure 1: i) learning from *similar tasks*; and ii) learning from *distant tasks*. For example, when we aim at training a text classifier over the sentiment classification dataset SST-2 [15], a similar task can be the tasks of the same type, such as MR [16] and CR [17]. In contrast, when similar datasets are unavailable for us (which is a more challenging setting), it is also beneficial to leverage training data from distant NLP tasks to improve the model for SST-2, such as the Natural Language Inference (NLI) datasets MNLI [18] and SNLI [19].

To accommodate both learning settings, *TransPrompt v2* employs a unified learning paradigm, with the high-level architecture shown in Figure 2. *TransPrompt v2* firstly employs a *Multi-task Meta-knowledge Acquisition* (MMA) procedure to learn the transferable representations of prompt encoders across multiple tasks. For learning across similar tasks, *TransPrompt v2* leverages task-specific prompt encoders to capture the private knowledge of each

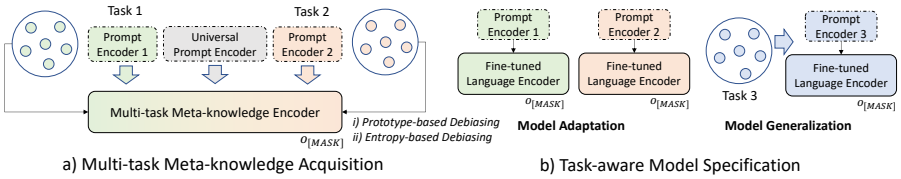


Fig. 2 The high-level architecture of the *TransPrompt v2* framework. In the toy example, Task 1 and Task 2 are existing tasks, while Task 3 is a new task for the meta-learner to generalize. (Best viewed in color.)

task, while a universal prompt encoder is exploited to learn the shared semantics across tasks. For learning across distant tasks, we additionally introduce task type descriptions with pseudo templates and then propose intra-type and inter-type prompt encoders to make PLMs better mine implicit relations among multiple distant tasks. To reduce over-fitting and make the underlying PLM more task-agnostic and less unbiased towards any specific tasks, we further propose two *de-biasing* techniques, namely *prototype-based de-biasing* and *entropy-based de-biasing*. The learned model can be viewed as a *meta-learner* for a group of NLP tasks.

After MMA, *TransPrompt v2* takes the *Task-aware Model Specification* (TMS) step, which can be further divided into two cases. i) When the model is adapted to existing tasks during MMA, a variation of P-tuning [8] can be applied for effective adaptation. ii) When it is required to fit a previously unseen task, a *model generalization* strategy is employed, specifically considering the universal prompting knowledge in the model. This is often the case where re-training of the *meta-leaner* across all the tasks is infeasible due to data privacy or computation efficiency issues.

For evaluation, we test the *TransPrompt v2* framework on three sets of few-shot NLP tasks (including seven public datasets in total): i) sentiment analysis; ii) NLI; and iii) paraphrase. Experimental results show that *TransPrompt v2* consistently outperforms both single-task and cross-task strong baselines. We further show that i) the meta-learner trained by *TransPrompt v2* is effective to generalize to unseen tasks; and ii) *TransPrompt v2* also outperforms popular fine-tuning algorithms when learning with full training sets.

In summary, the major contributions of this work are:

- We introduce the novel *TransPrompt v2* framework to learn cross-task transferable knowledge for prompt-based few-shot text classification.
- A prompt-based meta-learner training algorithm with two de-biasing techniques is presented to capture transferable knowledge. Specifically, the extended *TransPrompt v2* framework can be employed to learn from either similar NLP tasks or distant NLP tasks.
- Experiments on multiple types of NLP tasks show that the proposed *TransPrompt v2* framework consistently outperforms strong baselines for both few-shot learning and standard fine-tuning.

The rest of this paper is organized as follows. Section 2 summarizes the related work. Details of *TransPrompt v2* and experimental results are presented in Section 3 and Section 4. Finally, we discuss the limitations and extensions of our work in Section 5, and draw the conclusion in Section 6.

2 Related Work

In this section, we summarize the related work on PLMs, PLM prompting, transfer learning and meta-learning.

2.1 Pre-trained Language Models

With large-scale pre-training, PLMs have achieved significant improvements over various NLP tasks through simple task-specific fine-tuning stages. ELMo [20] is one of the early researches to pre-train contextual representations over large-scale unstructured corpora. BERT [2] captures contextual semantic representations by multi-layer transformer encoders [21], and pre-trained with two self-supervised learning objectives, i.e., Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Recently, a series of transformer encoder-based methods are proposed to improve the performance of the vanilla BERT model, including RoBERTa [3], ALBERT [4], XLNet [5], StructBERT [22], SpanBERT [23], DeBERTa [24], etc. Additionally, structured knowledge graphs can also be used to enhance PLMs. Specifically, ERNIE-Baidu [25] and WKLM [26] employ entity-level masking and replacement strategies to enhance the representations of tokens by learning from the explicit knowledge in knowledge graphs. K-BERT [27] and CoLAKE [28] directly concatenate external triples with original texts, and convert them into a graph. ERNIE-THU [29] and KEPLER [30] integrate the pre-trained knowledge base embeddings (KBE) into PLMs through well-designed attentive fusion modules to improve the contextual representations. Similar works also include [31, 32] and others. Rather than using transformer encoders only, the encoder-decoder architecture and the decoder-only architecture have been employed to make PLMs better adapt to both natural language understanding and generation tasks, such as BART [33], T5 [34], GPT-3 [9], etc. As the neural architecture design of PLMs is not our focus, we do not further elaborate.

2.2 Prompt Learning for PLMs

For text classification tasks, directly fine-tuning the `[CLS]` head over PLMs may perform poorly in low-resourced settings [35]. Recently, the huge GPT-3 model [9] has been proposed to enable in-context learning by handcrafted templates and demonstrations. To apply in-context learning to BERT-style models, Schick and Schütze [12] introduce handcrafted Pattern-Verbalizer-Pair (PVP) to unify all text classification into cloze-style formats. To facilitate the automatic PVP generation, Gao et al. [7] present LM-BFF to generate discrete templates and label word mappings [34]. Other works [11, 36, 37] mine

prompts from the training corpus based on heuristic rules/semantic relations. However, these methods are time-consuming for mining optimized prompts for target tasks. A series of methods are proposed to learn continuous/soft prompt embeddings, such as P-tuning [8], P-tuning-V2 [38], OptiPrompt [39], Prefix-tuning [40]. Zhao et al. [41] and Gu et al. [42] focus on the hybrid training with both discrete and continuous prompts. He et al. [43] consider the automatic expansion of label words and presents Knowledgeable Prompt-tuning (KPT) to utilize knowledge for the construction of verbalizers. The following two methods [44, 45] prompt the PLMs to make language inference in zero-shot learning. In addition, a few works [46–48] focus on prompts for specific NLP tasks, such as sentiment analysis and information extraction. We further leverage data from non-target NLP tasks to make prompt-tuned PLMs have better capacities of adapting to unseen NLP tasks.

2.3 Transfer Learning and Meta-learning

Transfer learning aims to transfer knowledge or resources from source domains to target domains [49–52]. For deep neural networks, it is common practice to learn similar tasks by multi-task learning [53]. With the popularity of PLMs, fine-tuning has become the standard practice by learning from PLMs for similar tasks [54, 55]. In contrast, meta-learning aims to learn models that can quickly adapt to different tasks with little training data available [56, 57], typically formulated as an *N-way K-shot* problem. Meta-learning algorithms have been applied in few-shot NLP tasks, such as text classification [58], relation extraction [59], question answering [60] and knowledge base completion [61]. We notice that our proposed *TransPrompt v2* framework is not a typical *N-way K-shot* paradigm. Similar to [13, 62, 63], *TransPrompt v2* can be viewed as a combination of transfer learning and meta-learning, which learns transferable knowledge from similar and distant tasks to improve the performance of few-shot text classification, either for existing tasks or new tasks.

3 The *TransPrompt v2* Framework

In this section, we formally present our task. After that, the techniques of the proposed *TransPrompt v2* framework are introduced in detail.

3.1 Overview

3.1.1 Task Description

As *TransPrompt v2* can handle transfer few-shot learning across both similar and distant tasks. We begin with a summary of our task when only similar tasks are considered.

Let $\mathcal{T}_1, \cdots, \mathcal{T}_M$ be M few-shot text classification tasks. The m-th task can be formulated as: $\mathcal{T}_m : x \rightarrow y$, where x and $y \in \mathcal{Y}$ represent the input text and the classification label, respectively. $\mathcal{Y}$ is the pre-defined label set with $|\mathcal{Y}| = N$, where N is a pre-defined constant. In our setting, we assume that

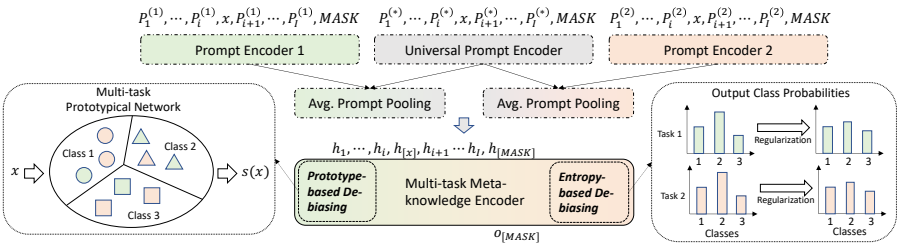


Fig. 3 The model architecture of the meta-learner training process during multi-task learning across similar tasks. For simplicity in this figure, we assume there are two tasks and three classes for few-shot text classification. The multi-domain few-shot datasets are passed through task-specific and universal prompt encoders. After that, a multi-task meta-knowledge encoder is employed to capture the transferable knowledge across tasks, with two de-biasing techniques proposed. (Best viewed in color.)

there are K training samples associated with each class $y \in \mathcal{Y}$ in each task $\mathcal{T}_m$. Hence, we have a training set $\mathcal{D}_m$ for each task $\mathcal{T}_m$, each containing $N \times K$ samples. The total number of training instances of M tasks is $N \times K \times M$. [1]

When we aim at learning across distant tasks, it is hard to ensure that all the tasks share the same label set. Hence, we have m-th task: $\mathcal{T}_m : x \rightarrow y$ with $y \in \mathcal{Y}_m$ where $\mathcal{Y}_m$ is the classification label set of the task $\mathcal{T}_m$. Similar to the previous case, we assume that there are K training samples associated with each class. Therefore, the number of the training samples in $\mathcal{D}_m$ can be calculated as: $K \cdot |\mathcal{Y}_m|$. Hence, the total number of training instances of M tasks is $K \cdot \sum_m |\mathcal{Y}_m|$.

3.1.2 Model Overview

In *TransPrompt v2*, we train a *meta-learner* $\mathcal{F}_{meta}$ with parameters initialized from any PLMs, based on the M few-shot training sets $\mathcal{D}_1, \cdots, \mathcal{D}_M$. As mentioned above, we consider two scenarios. 1) Training across similar tasks. In this case, the selected M training tasks share the same label space. 2) Training across distant tasks. The selected M training sets can have different label spaces. This means, there exist at least two tasks $\mathcal{T}_i$ and $\mathcal{T}_j$ in $\mathcal{T}_1, \cdots, \mathcal{T}_M$ such that $\mathcal{Y}_i \neq \mathcal{Y}_j$.

After training the meta-learner, $\mathcal{F}_{meta}$ is adapted to each task $\mathcal{T}_m$ based on its own training set $\mathcal{D}_m$. The task-specific model is denoted as $\mathcal{F}_m$. As $\mathcal{F}_{meta}$ is designed to digest the *transferable knowledge* across tasks, rather than simple multi-task learning, $\mathcal{F}_{meta}$ can also be adapted to previously unseen tasks. Due to the data privacy or computation efficiency issues, when the few-shot training set $\tilde{\mathcal{D}}$ of a similar task $\tilde{\mathcal{T}}$ is not available during the training process of $\mathcal{F}_{meta}$, we explore how *TransPrompt v2* can be used to generate an accurate model $\tilde{\mathcal{F}}$ based on $\mathcal{F}_{meta}$ and $\tilde{\mathcal{D}}$. In this case, $\mathcal{F}_{meta}$ does not have any knowledge of the new task $\tilde{\mathcal{T}}$ when in the cross-task training stage.

[1] Note that the input x can be either a single sentence or a sentence pair (which has the same setting as that of BERT [2]). For simplicity, we uniformly denote the input as x throughout this paper.

In the following, we introduce the detailed techniques of the *TransPrompt v2* framework. We elaborate on the two major stages of our framework, considering both similar and distant NLP tasks, i.e., *Multi-task Meta-knowledge Acquisition* (MMA) and *Task-aware Model Specification* (TMS). After that, we discuss how to apply *TransPrompt v2* to standard fine-tuning scenarios where we have relatively large training sets, instead of solving the *N-way K-shot* problem.

3.2 Prompt Encoding for Learning across Similar Tasks

We first consider the situation where *TransPrompt v2* aims at transfer prompt learning across similar tasks. For clarity, the general architecture of the meta-learner for similar-task learning is illustrated in Figure 3. For prompt encoding, we augment an input sentence x with some pseudo template tokens. The prompt template $t(x)$ is shown as follows:

$$t(x) = P_1, \cdots, P_i, x, P_{i+1}, \cdots, P_I, MASK,$$

where P_i denotes the prompt pseudo token (as proposed in [8]), I is the total number of pseudo tokens, and $MASK$ is a special token as the placeholder for model output. To learn the dependency between pseudo tokens and improve the model performance, following P-tuning [8], we employ a prompt encoder $\mathrm{PE}(x)$ containing one bidirectional LSTM network with multi-layer perceptrons (MLP) to capture the dependency between each pseudo token (this means we do not compute on original texts x and the masked token). Formally, we have:

$$\mathrm{PE}_\phi(x) = \mathrm{MLP}_\phi\big(\mathrm{BiLSTM}_\phi(t(x)\big),$$

where ϕ denotes trainable parameters of the prompt encoder.

As the *TransPrompt v2* framework is placed in the multi-task setting, for each task $\mathcal{T}_m$, we have a task-specific prompt template $t^{(m)}(x)$ as follows:

$$t^{(m)}(x) = P_1^{(m)}, \cdots, P_i^{(m)}, x, P_{i+1}^{(m)}, \cdots, P_I^{(m)}, MASK.$$

Besides, we also define a universal prompt template $t^{(*)}(x)$ for all the similar tasks:

$$t^{(*)}(x) = P_1^{(*)}, \cdots, P_i^{(*)}, x, P_{i+1}^{(*)}, \cdots, P_I^{(*)}, MASK.$$

For an instance $(x, y) \in \mathcal{D}_m$, the prompt embedding $\mathbf{H}^m(x)$ can be computed as follows:

$$\mathbf{H}^m(x) = \mathrm{AvgPool}\big(\mathrm{PE}_{\phi^m}(t^{(m)}(x)), \mathrm{PE}_{\phi^*}(t^{(*)}(x))\big),$$

where ϕ^m and ϕ^* denote the parameters of task-specific and universal prompt encoders, respectively. The average pooled results from both task-specific and

universal prompt encoders are treated as the prompt embedding $\mathbf{H}^m(x)$. The prompt embedding is a sequence as the input of the PLM, i.e.,

$$h_1, \cdots, h_i, h_{[x]}, h_{i+1}, \cdots, h_I, h_{[MASK]}$$

where $h_{[x]}$ is the sequence embedding of input x, and $h_{[MASK]}$ is the masked output token embedding. As the parameters of the prompt encoders are fully differentiable, during backpropagation, they can effectively capture the task-specific and universal knowledge.

3.3 Prompt Encoding for Learning across Distant Tasks

By intuition, PLMs trained over tasks with similar label spaces can learn the transferable knowledge across these tasks. However, it is difficult for the previous prompt encoding approach to directly apply to various tasks with different label spaces. In addition, the huge difference in the intrinsic semantic distributions among these distant tasks makes PLMs difficult to learn the implicit mutual relations. Hence, we extend our prompt encoding method to distant NLP tasks.

Suppose that our task is to improve the model performance over a sentiment analysis task SST-2 [15], and we only have two other NLI datasets MNLI [18] and SNLI [19]. Thus, we have $R = 2$ type of tasks and $M = 3$ datasets during the MMA stage. To make PLMs understand the task semantics more clearly, we design concise text descriptions for each task type and concatenate them with the corresponding prompts. Specifically, given one task $\mathcal{T}_m$ of r-th type. For each instance $(x, y) \in \mathcal{D}_m$, the task-specific and universal prompt templates are denoted as $t^{(m)}(x)$ and $t^{(*)}(x)$, which is the same as described previously. We also provide the prompt template with the task type description denoted as $t^{(r)}(x)$ where instances with the same task type share the same template, represented as:

$$Q_1^{(r)}, \cdots, Q_{I'}^{(r)}, P_1^{(r)}, \cdots, P_i^{(r)}, x, P_{i+1}^{(r)}, \cdots, P_I^{(r)}, MASK,$$

where $Q_i^{(r)}$ refers to the text tokens of the description, I' denotes the length of the description. The description tokens are the prefix of pseudo template.

Different from the previous approach, we consider two aspects of prompt encoding, include the intra-type prompt embedding $\mathbf{H}^m_{intra}(x)$ and the inter-type prompt embedding $\mathbf{H}^m_{inter}(x)$, defined as follows:

$$\mathbf{H}^m_{intra}(x) = \text{AvgPool}\big(\text{PE}_{\phi^m}(t^{(m)}(x)), \text{PE}_{\phi^r}(t^{(r)}(x))\big),$$

$$\mathbf{H}^m_{inter}(x) = \text{AvgPool}\big(\text{PE}_{\phi^r}(t^{(r)}(x)), \text{PE}_{\phi^*}(t^{(*)}(x))\big),$$

where ϕ^r and ϕ^* denote the parameters of type-specific and universal prompt encoders, respectively.

In this manner, for one batch of examples, the parameters of the type-specific prompt encoder $\text{PE}_{\phi^r}(\cdot)$ can be optimized by all the data of the r-th type, while the task-specific prompt encoder $\text{PE}_{\phi^m}(\cdot)$ only focuses on the task $\mathcal{T}_m$. At the same time, the universal prompt encoder $\text{PE}_{\phi^*}(\cdot)$ aims to learn the universal knowledge from all examples. Hence, $\text{PE}_{\phi^m}(\cdot)$ and $\text{PE}_{\phi^r}(\cdot)$ can learn the mutual relations among data with the same type r and obtains representations $\mathbf{H}^m_{intra}(x)$, while $\text{PE}_{\phi^r}(\cdot)$ and $\text{PE}_{\phi^*}(\cdot)$ jointly learn the semantics of different task types, and generate representations $\mathbf{H}^m_{inter}(x)$. To prevent introducing too many additional parameters, the parameters of all prompt encoders in the intra-type setting are shared. Finally, we directly combine the two prompt embeddings by:

$$\mathbf{H}^m(x) = \alpha \mathbf{H}^m_{intra}(x) + (1-\alpha)\mathbf{H}^m_{inter}(x),$$

where α is the balancing coefficient, with $0 < \alpha < 1$.

3.4 The Training Process of the Meta-learner

During model pre-training, we regard the prompt embedding $\mathbf{H}^m(x)$ as the model input and train the meta-learner via multi-task learning over the MLM head. As large PLMs can easily suffer from over-fitting during few-shot learning [7], in cross-task scenarios, the meta-learner would unfortunately memorize the *non-transferable knowledge* from non-target tasks. To alleviate this problem, we propose two de-biasing techniques to obtain a more *unbiased* meta-learner encoded with *transferable knowledge*, namely i) *prototype-based de-biasing* and ii) *entropy-based de-biasing*.

Prototype-based De-biasing. This technique aims to give more importance to *prototypical instances across tasks* during the training process of the meta-learner. Here, we extend [64] to construct a lite *Multi-task Prototypical Network*. In the network, the task-specific class centroid embedding $\mathbf{c}_m(y)$ ($y \in \mathcal{Y}$ for similar tasks and $y \in \mathcal{Y}_m$ for distant task) w.r.t. the task $\mathcal{T}_m$ is computed and stored as:

$$\mathbf{c}_m(y) = \frac{1}{|\mathcal{D}_{m,y}|} \sum_{(x,y)\in\mathcal{D}_{m,y}} \mathcal{E}(x)$$

where $\mathcal{D}_{m,y}$ is the subset of $\mathcal{D}_m$ such that each instance in $\mathcal{D}_{m,y}$ has the label y, and $\mathcal{E}(x)$ is the representation of x generated by the meta-learner described previously [2]. For each instance $(x, y) \in \mathcal{D}_m$, we pass the text x through the network to generate the *cross-task prototype score*, denoted as

[2]In this work, we use the pooled results of the last layer of the PLM encoder as $\mathcal{E}(x)$.

$s(x)$. For learning across similar tasks, we have:

$$s(x) = \zeta \cdot \frac{\text{sim}(\mathcal{E}(x), \mathbf{c}_m(y))}{\sum_{\tilde{y}\in\mathcal{Y}} \text{sim}(\mathcal{E}(x), \mathbf{c}_m(\tilde{y}))} + \frac{1-\zeta}{M-1} \sum_{\tilde{m}=1(m\neq\tilde{m})}^{M} \frac{\text{sim}(\mathcal{E}(x), \mathbf{c}_{\tilde{m}}(y))}{\sum_{\tilde{y}\in\mathcal{Y}} \text{sim}(\mathcal{E}(x), \mathbf{c}_{\tilde{m}}(\tilde{y}))},$$

where $0 < \zeta < 1$ is a pre-defined balancing factor, and $\text{sim}(\cdot, \cdot)$ is the similarity function between two embeddings. We can see that an instance receives a higher score if it is semantically related to the centroids from both the task $\mathcal{T}_m$ itself and other tasks, hence is *more transferable across tasks*. For learning across distant tasks, considering the label space differences, it is also straightforward to compute the score $s(x)$ as follows:

$$s(x) = \zeta \cdot \frac{\text{sim}(\mathcal{E}(x), \mathbf{c}_m(y))}{\sum_{\tilde{y}\in\mathcal{Y}_m} \text{sim}(\mathcal{E}(x), \mathbf{c}_m(\tilde{y}))} + \frac{1-\zeta}{M-1} \sum_{\tilde{m}=1(m\neq\tilde{m})}^{M} \frac{\text{sim}(\mathcal{E}(x), \mathbf{c}_{\tilde{m}}(y))}{\sum_{\tilde{y}\in\mathcal{Y}_{\tilde{m}}} \text{sim}(\mathcal{E}(x), \mathbf{c}_{\tilde{m}}(\tilde{y}))}.$$

By treating $s(x)$ as the optimization weight, the overall loss function $\mathcal{L}(\Theta)$ of $\mathcal{F}_{meta}$ can be given by:

$$\mathcal{L}(\Theta) = \sum_{m=1}^{M} \sum_{(x,y)\in\mathcal{D}_m} s(x) f(x, y;\ \Theta) + \lambda_1 \|\Theta\|,$$

where Θ is the collection of all model parameters, $f(x, y;\ \Theta)$ is the *sample-wise cross-entropy loss*, and λ_1 is the regularization hyper-parameter.

Entropy-based De-biasing. One potential risk of applying the *prototype-based de-biasing* technique only is obtaining a *non task-agnostic* meta-learner. Consider three tasks $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$. If $\mathcal{T}_1$ and $\mathcal{T}_2$ are highly similar, and $\mathcal{T}_3$ is more dissimilar. Instances in $\mathcal{D}_1$ and $\mathcal{D}_2$ would naturally receive high prototype scores, making the meta-learner *biased* towards $\mathcal{T}_1$ and $\mathcal{T}_2$, and pays little attention to $\mathcal{T}_3$. Hence, when the meta-learner is required to fit $\mathcal{T}_3$, it may have poor parameter initialization settings. To make it *more task-agnostic*, inspired by [65], we consider the model prediction entropy $\mathcal{H}(\mathcal{D}_m)$ over $\mathcal{D}_m$:

$$\mathcal{H}(\mathcal{D}_m) = -\frac{1}{|\mathcal{D}_m|} \sum_{(x,y)\in\mathcal{D}_m} \sum_{\hat{y}\in\mathcal{Y}_*} \hat{y}(x) \log \hat{y}(x),$$

where $\hat{y}(x)$ is the predicted probability of x being assigned to the class $\hat{y} \in \mathcal{Y}$. If we learn across similar tasks, we have $\mathcal{Y}_* = \mathcal{Y}$; otherwise, $\mathcal{Y}_* = \mathcal{Y}_m$. We can see that when $\mathcal{H}(\mathcal{D}_m)$ is used as a part of the model regularizers, the meta-learner will be *less over-trained* on any specific tasks.

By plugging the term $\mathcal{H}(\mathcal{D}_m)$ into the loss function $\mathcal{L}(\Theta)$, we have the new loss function $\mathcal{L}^{'}(\Theta)$:

$$\mathcal{L}^{'}(\Theta) = \sum_{m=1}^{M} \sum_{(x,y)\in\mathcal{D}_m} (s(x) f(x, y;\ \Theta) - \frac{\lambda_2}{|\mathcal{D}_m|} \sum_{\hat{y}\in\mathcal{Y}_*} \hat{y}(x) \log \hat{y}(x)) + \lambda_1 \|\Theta\|,$$

where λ_2 is the regularization hyper-parameter.

Optimization Procedure. Despite its simple formula, minimizing $\mathcal{L}^{'}(\Theta)$ is a non-trivial problem. This is because when we calculate $s(x)$, we must obtain model parameters of the PLM beforehand, which is not available before the training process. On the other hand, the optimization of $\mathcal{L}^{'}(\Theta)$ requires the values of $s(x)$ for all training samples, which poses the *"chicken-and-egg"* problem.

We employ a *dual optimization* process to solve the problem of $\mathcal{L}^{'}(\Theta)$. In the initial stage, all $s(x)$s are uniformly initialized. Next, we fix $s(x)$ scores as constants to minimize $f(x, y;\ \Theta)$ in $\mathcal{L}^{'}(\Theta)$. An inference procedure on the PLM can be applied to obtain all $s(x)$ scores. This process iterates for a certain number of epochs. Readers can also refer to Algorithm 1 for an algorithmic overview.

3.5 Task-aware Model Specification

After MMA, the meta-learner can be adapted to specific tasks with ease. For a task $\mathcal{T}_m$ that has already "seen" by the meta-learner, we fine-tune the corresponding task-specific prompt encoder and the PLM by minimizing the loss function $\mathcal{L}^{(m)}(\Theta)$:

$$\mathcal{L}^{(m)}(\Theta) = \sum_{(x,y)\in\mathcal{D}_m} f(x, y;\ \Theta) + \lambda_1 \|\Theta\|,$$

which is a variant of P-tuning [8] with better parameter initialization.

For a previously unseen task $\tilde{\mathcal{T}}$, the *model generalization* strategy is employed. Here, we use the universal prompt encoder to initialize its prompt encoder. The entire model is trained over the dataset $\tilde{\mathcal{D}}$, with the loss function $\tilde{\mathcal{L}}(\Theta)$ defined as follows:

$$\tilde{\mathcal{L}}(\Theta) = \sum_{(x,y)\in\tilde{\mathcal{D}}} f(x, y;\ \Theta) + \lambda_1 \|\Theta\|,$$

As the meta-learner is highly generalized, it can provide good initialization for the few-shot learning task $\tilde{\mathcal{T}}$.

Algorithm 1 Meta-learner Training Algorithm

for each instance $(x, y) \in \bigcup_{m=1}^{M} \mathcal{D}_m$ **do**
    Uniformly set $s(x) = 1$;
end for
while number of training epochs does not reach a limit **do**
    while current training epoch is not finished **do**
        Sample a batch $\mathcal{B} = \{(x, y)\}$ from $\bigcup_{m=1}^{M} \mathcal{D}_m$;
        Use $\mathcal{B}$ to update $\mathcal{F}_{meta}$ by minimizing $\mathcal{L}^{'}(\Theta)$;
    end while
    for each instance $(x, y) \in \bigcup_{m=1}^{M} \mathcal{D}_m$ **do**
        Compute $s(x)$ based on the updated model;
    end for
end while
return the meta-learner $\mathcal{F}_{meta}$ (i.e., parameters of the PLM and multiple prompt encoders).

3.6 Learning with Full Training Sets

TransPrompt v2 can also be applied for standard fine-tuning when we have relatively large training sets with few modifications. During MMA, we notice that when it is not an *N-way K-shot* problem, the sizes of $\mathcal{D}_1, \cdots, \mathcal{D}_M$ can be significantly different. Optimizing $\mathcal{L}^{'}(\Theta)$ directly on these datasets would make the meta-learner *biased towards large datasets*. To remedy this dilemma, when we sample a batch from $\mathcal{D}_1, \cdots, \mathcal{D}_M$, instead of randomly selection, we employ stratified sampling where training instances are selected with the probability proportional to the dataset distribution $\Pr(\mathcal{D}_m)$:

$$\Pr(\mathcal{D}_m) = \frac{\log |\mathcal{D}_m| + \gamma}{\sum_{\tilde{m}=1}^{M} \log |\mathcal{D}_{\tilde{m}}| + \gamma},$$

where $\gamma > 0$ is a smoothing factor. This results in the *over-sampling* of small datasets and the *under-sampling* of large datasets, which helps to improve the generalization capability of the model.

4 Experiments

In this section, we conduct extensive experiments to evaluate the *TransPrompt v2* framework and compare it against strong baselines.

4.1 Datasets and Experimental Settings

Following [1, 7, 8], we select seven public datasets to evaluate *TransPrompt v2*, divided into three groups of NLP tasks: sentiment analysis (SST-2 [15], MR [16] and CR [17]), NLI (MNLI [18] and SNLI [19]) and paraphrase (MRPC [66] and

Table 1 The specific information of each dataset. `[<s1>]` and `[<s2>]` are the original sentences, `[MASK]` is the maked token, and `[p]` is the pseudo token.

Task Type	Task	#Train	#Test	Template Format	Description
Sentiment	SST-2	6,920	872	`[<s1>]. [p1] ⋯ [MASK].`	A sentiment analysis task.
	MR	8,662	2,000	`[<s1>]. [p1] ⋯ [MASK].`	
	CR	1,775	2,000	`[<s1>]. [p1] ⋯ [MASK].`	
NLI	MNLI	392,702	9,815	`[<s1>]. [p1] ⋯ [MASK]. [<s2>].`	A natural language inference.
	SNLI	549,367	9,842	`[<s1>]. [p1] ⋯ [MASK]. [<s2>].`	
Paraphrase	MRPC	3,668	408	`[<s1>]. [p1] ⋯ [MASK]. [<s2>].`	A paraphrase task.
	QQP	363,846	40,431	`[<s1>]. [p1] ⋯ [MASK]. [<s2>].`	

QQP[3]). Note that we do not use other benchmark datasets (such as CoLA) because datasets on similar tasks to CoLA are not available to us. Hence, it is impossible to train the meta-learner in *TransPrompt v2*. We provide the template format and task type description for each task. For the verbalizer, we follow [1, 7] to use the top-3 label mapping and report the average results in experiments. The detailed information of these datasets is reported in Table 1. The training/development/testing splits are the same as [1].

For few-shot learning, the evaluation protocols are the same as [1, 7]. The underlying PLM is the RoBERTa large model (with 335M parameters) [3] and we set $K = 16$. We measure the average performance in terms of accuracy across 5 different randomly sampled training and development splits. Refer to [7] for more experimental settings. We employ the following methods as single-task baselines:

- Standard BERT-style [2] fine-tuning[4];
- The LM-BFF prompting model [7] (with both manually-compiled and automatically-mined prompts)[5];
- P-tuning [8], which produces state-of-the-art performance for few-shot learning based on continuous prompt embeddings[6].

Because we focus on learning transferable prompting knowledge across tasks, we also use the multi-task version baselines of BERT fine-tuning [54], LM-BFF [7] and P-tuning [8] for learning across both similar and distant tasks. Specifically, we employ separate prompts (either discrete prompts or continuous prompt embeddings) for different tasks in the multi-task versions of LM-BFF and P-tuning. In addition, Meta Fine-tuning (MFT) [13] is employed as a baseline for similar tasks, while the Meta Distant Transfer Learning framework (Meta-DTL) [14] is used as a baseline for learning across distant tasks. During MMA from similar tasks, we constrain that the knowledge is transferred within the same group of NLP tasks. For example, the meta-learner for sentiment analysis is jointly trained over the few-shot training sets of SST-2, MR, and CR. When training across distant tasks, apart from the target task

[3] https://www.quora.com/q/quoradata/.
[4] https://github.com/huggingface/transformers.
[5] https://github.com/princeton-nlp/LM-BFF.
[6] https://github.com/THUDM/P-tuning.

Table 2 The few-shot testing results of *TransPrompt v2* and baselines in terms of accuracy (%). "man", "auto" and "mtl" refer to manually-compiled prompts, automatically-mined prompts, and multi-task learning, respectively. * refers to the multi-task variants of the original approaches. Hereinafter the same.

Method	Task: Sentiment Analysis			Task: NLI		Task: Paraphrase		Average
	SST-2	MR	CR	MNLI	SNLI	MRPC	QQP	
Single-task Methods								
Fine-tuning [2]	81.42	76.15	84.50	54.17	44.45	73.28	59.64	67.66
LM-BFF (man) [7]	90.75	86.60	90.50	63.62	70.77	74.05	60.27	76.65
LM-BFF (auto) [7]	91.62	87.25	91.80	64.25	71.21	74.23	60.59	77.28
P-tuning [8]	91.85	86.60	91.75	62.41	70.28	66.42	60.57	75.70
Cross-task Methods w. Learning from Similar Tasks								
Fine-tuning (mtl) [54]	83.37	79.30	84.75	41.32	48.14	53.12	59.31	64.19
Meta Fine-tuing [13]	86.32	83.85	88.42	48.52	58.20	71.56	67.12	72.00
LM-BFF (mtl) [7]*	91.97	87.45	90.70	69.09	75.90	72.18	67.40	79.24
P-tuning (mtl) [8]*	93.12	87.75	91.35	68.83	74.24	70.83	69.99	79.44
TransPrompt [1]	**93.58**	**88.80**	**92.00**	**71.90**	**76.99**	**75.98**	**75.80**	**82.15**
Cross-task Methods w. Learning from Distant Tasks								
Meta-DTL [14]	82.15	77.23	84.52	52.35	53.42	69.82	65.42	69.27
LM-BFF (mtl) [7]*	90.63	85.65	89.95	63.37	**73.34**	70.39	67.85	77.43
P-tuning (mtl) [8]*	91.05	86.00	90.30	63.49	72.12	71.07	70.99	77.86
TransPrompt [1]	90.47	86.05	90.30	62.56	71.97	71.83	70.17	77.62
TransPrompt v2	**91.97**	**87.60**	**91.05**	**65.58**	72.78	**73.75**	**71.47**	**79.53**

(e.g., SST-2), we also select other types of datasets (e.g., NLI and paraphrase) for multi-task learning. After multi-task training, only training data of the target task is used to perform task-specific prompt-tuning.

For fair comparison, we re-produce all baselines based on their open-source codes under the same settings. Our own *TransPrompt v2* algorithm is implemented in PyTorch and run with NVIDIA V100 GPUs. In default, we set $\zeta = \alpha = 0.5$, $\gamma = 0.001$ and $\lambda_2 = 0.01$. The parameter regularizers are the same as in [8]. The model is trained with the Adam optimizer [67]. The batch size is set as 6 and 16 in few-shot and full scenarios, respectively. The model architecture of prompt encoders is the same as [8]. Therefore, the increased number of parameters of *TransPrompt v2* remains minimal. We further tune the learning rates, epochs, and the length of pseudo tokens I, with results reported in the following experiments.

4.2 General Experimental Results

The results of *TransPrompt v2* and all baselines on all seven testing sets for few-shot learning are shown in Table 2. From the experimental results, we have drawn the following conclusions:

- Prompting baselines (such as LM-BFF and P-tuning) outperform the standard fine-tuning method by a large margin in both single-task learning and cross-task learning settings. This shows prompts are useful for few-shot learning.

- Based on our re-production, LM-BFF and P-tuning have similar performance. Yet, automatically-mined prompts are still sightly better than manually-compiled prompts for LM-BFF. The gains between LM-BFF and P-tuning over similar and distant cross-task learning are 0.20% and 0.43%, respectively, which shows that continuous prompt embeddings are more suitable for multi-task learning than discrete prompts.
- Comparing multi-task learning approaches to similar tasks, we find that all methods trained in distant scenarios are decreased over most of the datasets. The results clearly and intuitively reflect the phenomenon that there is a large difference between each type of task. Hence, this setting is significantly more challenging.
- As for MMA across distant tasks, the performance of our previous version *TransPrompt* [1] is similar to other prompt-based baselines, which indicates that existing prompt-based methods are hard to deal with a different types of tasks. Fortunately, we find that the performance can be improved by the utilization of external task type-specific prompting knowledge (i.e., task-type descriptions). Specifically, according to the overall averaged results in Table 2, *TransPrompt v2* outperforms all baselines, and the average improvement is around 2% in terms of accuracy, compared to the strongest baseline (i.e., the multi-task version of P-tuning).
- For fair comparison in cross-task learning, we also choose two state-of-the-art fine-tuned baselines named MFT and Meta-DTL, which dedicates to meta-learning over similar tasks and distant tasks, respectively. Results show that our method still performs better in both settings.
- Finally, we have conducted the *paired t-tests* over the results produced on all tasks. The results show that the improvement of *TransPrompt v2* is statically significant (with the p-value $p < 0.01$). This further demonstrates the effectiveness of the proposed *TransPrompt v2*.

In the following, we study how *TransPrompt v2* improves the performance in various aspects.

4.3 Ablation Studies

In the *TransPrompt v2* framework, we propose two de-biasing techniques to improve the effectiveness of the meta-learner, i.e, *prototype-based* and *entropy-based*. Here, we remove each one and all de-biasing techniques and implement the variants of *TransPrompt v2*. We conduct experiments over both similar and distant tasks. As shown in Table 3, both de-biasing techniques are proved effective for *TransPrompt v2*. Particularly, *prototype-based de-biasing* plays a slightly more important role than *entropy-based de-biasing* in most scenarios. We conclude that de-biasing the meta-learner is crucial for obtaining cross-task knowledge.

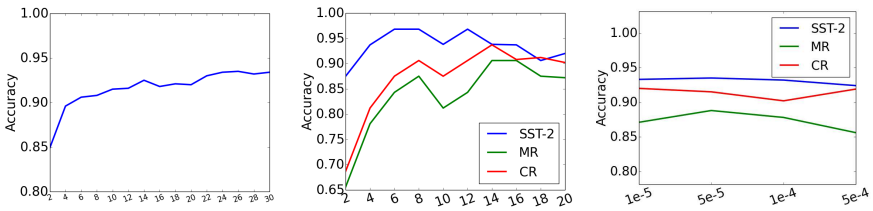
In addition, we also study how *TransPrompt v2* performs well in distant cross-task scenarios. We take SST-2, MR, and CR as examples, and test the performance when removing task type descriptions and corresponding prompt

Table 3 Ablation study of *TransPrompt v2* for few-shot learning. Experimental results are reported on the testing sets in terms of accuracy (%). **Similar** and **Distant** denote the similar and distant cross-task learning, respectively.

Task	w/o. Prototype-based De-biasing		w/o. Entropy-based De-biasing		w/o. Both		Full Implementation	
	Similar	Distant	Similar	Distant	Similar	Distant	Similar	Distant
SST-2	92.90	90.64	**93.18**	**91.28**	92.67	90.10	93.58	91.97
MR	87.97	86.30	**88.14**	**87.30**	87.75	86.05	88.80	87.60
CR	91.50	**90.55**	**91.70**	90.50	91.07	89.80	92.00	91.05
MNLI	67.72	63.77	**69.74**	**65.05**	67.08	63.86	71.90	65.58
SNLI	**76.76**	70.29	76.66	**71.15**	76.08	70.91	76.99	72.78
MRPC	75.98	72.07	**76.72**	**73.00**	68.38	65.70	75.98	73.75
QQP	73.44	**71.03**	**74.00**	70.98	73.02	68.40	75.80	71.47
Average	80.90	77.81	**81.45**	**78.47**	79.44	76.40	82.14	79.93

Table 4 The accuracy effectiveness (%) of the task type descriptions and corresponding intra/inter-type prompt embeddings.

Task	SST-2	MR	CR	Average
w/o. description	90.98	86.70	90.65	89.44
w/o. inter-type	90.64	86.20	90.45	89.10
w/o. intra-type	90.64	86.15	90.50	89.10
w/o. all	90.47	86.05	90.30	88.94
TransPrompt v2	**91.97**	**87.60**	**91.05**	**90.21**



(a) Tuning the learning epoch of the meta-learner during MMA.

(b) Tuning the learning epoch of the fine-tuned PLM during TMS.

(c) Tuning the learning rate.

Fig. 4 Parameter analysis of the learning epoch and the learning rate for learning across similar tasks. The underlying task group is sentiment analysis.

encoders. Results in Table 4 demonstrate that each of techniques is vital for cross-task distant learning.

4.4 Parameter Analysis

We further tune the learning epoch and the learning rate during the training process of *TransPrompt v2* and report the performance over the development sets. Consider the similar-task learning scenario. We illustrate the results over SST-2, MR, and CR, shown in Figure 4. We fix the learning rate to be 1e-5 and

Table 5 The accuracy efficiency (%) of the pseudo prompt length I. We conduct experiments on sentiment analysis in few-shot cross-task learning from similar tasks.

Task	SST-2	MR	CR
$I = 1$	**93.58**	88.05	91.70
$I = 2$	93.14	**88.25**	**91.85**
$I = 3$	92.67	87.85	91.05
$I = 4$	92.97	87.30	91.05

tune the learning epochs. Figure 4(a) shows that the performance of the meta-learner becomes stable over 20 epochs (which is tested on the combination of three developments sets for SST-2, MR, and CR). Figure 4(b) gives the results of the three tasks during TMS. We can see that *TransPrompt v2* only needs a small number of training steps to reach the best performance on development sets of multiple tasks. It implies that we can leverage an early stopping strategy during experiments. For learning rate selection, we fix the training epoch to 20 and tune the learning rate from 1e-5 to 5e-4. As seen in Figure 4(c), the learning rate should be set in the range of 1e-5 to 5e-5.

Except for the analysis of training parameters, we also explore the efficiency of the hyper-parameter in the model, i.e., the length of pseudo prompt tokens. We choose all tasks from sentiment analysis and conduct experiments in few-shot distant cross-task settings. Results in Table 5 suggest that the length of pseudo prompts can influence the model accuracy. Based on our results, we recommend setting $I \in \{1, 2\}$.

4.5 Model Generalization to New Unseen Tasks

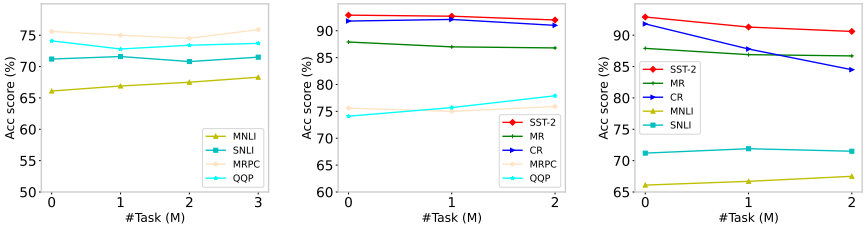
One advantage of *TransPrompt v2* is that it can train a meta-learner with *cross-task transferable knowledge* encoded. In this set of experiments, we consider two scenarios including generalization from similar tasks and distant tasks. All experiments are in few-shot settings.

For the generalization from similar tasks, we consider three tasks of sentiment analysis: SST-2, MR, and CR. Each time, we train the meta-leaner over two out of the three datasets (the MMA stage) and then make the model generalized to each of the other tasks (the TMS stage). In these settings, there are no training examples overlap between the two stages. For example, we train the meta-learner $\mathcal{F}_{meta}$ over the few-shot SST-2 and MR datasets and then take the TMS step over the few-shot CR dataset. Here, the meta-learner has no knowledge of CR before the TMS step. So, it can validate whether the model can solve new unseen tasks with knowledge of similar tasks. We test whether the usage of the meta-learner is better than simply applying LM-BFF [7] or P-tuning [8] initialized from PLMs. From Table 6, it is clearly reflected that the meta-learner brings improvements in all three cases, hence generalizing to new tasks accurately.

For the generalization of distant tasks, we explore how many distant tasks can make contributions to each target task. For example, we choose one group

Table 6 Model generalization results in terms of accuracy (%). LM-BFF and P-tuning are strong baselines w/o. the usage of the meta-learner.

Task	SST-2	MR	CR	Average
LM-BFF (m)	90.75	86.60	90.50	89.28
LM-BFF (a)	91.62	87.25	91.80	90.22
P-tuning	91.85	86.60	91.75	90.05
TransPrompt v2	**93.35**	**88.25**	**91.85**	**91.15**



(a) MMA training across $M(M \in \{0, 1, 2, 3\})$ tasks from sentiment analysis, and fine-tuning over MNLI, SNLI, MRPC and QQP.

(b) MMA training across $M(M \in \{0, 1, 2\})$ tasks from NLI, and fine-tuning over SST-2, MR, CR, MRPC and QQP.

(c) MMA training across $M(M \in \{0, 1, 2\})$ tasks from paraphrase, and fine-tuning over SST-2, MR, CR, MNLI and SNLI.

Fig. 5 Model generalization efficiency in distant cross-task settings. For each figure, the horizontal axis denotes the number M of tasks from the corresponding task group for training the meta-learner $\mathcal{F}_{meta}$, while the vertical axis represents the accuracy of each target unseen distant task $\mathcal{T}^*$ in the TMS stage. $M = 0$ is a special situation that represents having the TMS stage only.

of tasks (e.g., SST-2, MR, and CR in sentiment analysis.), and randomly select M tasks from this group for training the meta-learner $\mathcal{F}_{meta}$. Then, we make the model generalized to each of the other distant tasks (i.e., MNLI, SNLI, MRPC, and QQP). In this setting, the model can not see any data from the target task during MMA, which can be better to test the effectiveness of model generalization. As shown in Figure 5, we draw some conclusions:

- The overall results show that our method is able to fuse semantics knowledge from distant tasks to enhance the effectiveness of most of datasets.
- Tasks from sentiment analysis have a slight improvement for NLI and paraphrase, while the improvement is not obvious. As shown in Figure 5(a), when meta training overall sentiment tasks, only the MNLI task increases more than 1.0%.
- It is difficult for the task form sentiment analysis to capture transferable knowledge from tasks of NLI and paraphrase. Especially, when we train $\mathcal{F}_{meta}$ over tasks from paraphrase, the performance of sentiment analysis

Table 7 The testing results of *TransPrompt v2* and baselines with full training sets in terms of accuracy (%).

Method	Task: Sentiment Analysis			Task: NLI		Task: Paraphrase		Average
	SST-2	MR	CR	MNLI	SNLI	MRPC	QQP	
Single-task Baselines								
Fine-tuning [2]	93.00	90.15	90.90	82.87	87.87	72.28	89.53	86.65
LM-BFF (man) [7]	93.65	88.50	90.98	87.23	91.10	88.75	85.12	89.33
LM-BFF (auto) [7]	93.81	88.75	91.25	87.01	91.51	**88.97**	83.12	89.20
P-tuning [8]	93.69	90.10	90.25	87.17	91.67	**88.97**	90.87	90.38
Cross-task Baselines w. Learning from Similar Tasks								
Fine-tuning (mtl) [54]	94.72	90.65	91.05	87.10	91.80	69.85	90.20	87.91
Meta Fine-tuing [13]	95.70	91.25	91.42	83.67	89.48	78.92	89.72	88.59
LM-BFF (mtl) [7]*	95.41	90.45	91.50	86.76	88.25	69.36	90.32	87.43
P-tuning (mtl) [8]*	95.30	90.40	90.08	86.97	91.48	68.87	90.59	87.67
TransPrompt [1]	**96.05**	**91.78**	**91.59**	**88.70**	**91.88**	86.87	**91.27**	**91.16**
Cross-task Methods w. Learning from Distant Tasks								
Meta-DTL [14]	93.70	90.20	90.14	82.65	88.10	74.86	88.84	86.92
LM-BFF (mtl) [7]*	94.10	90.65	91.30	86.98	90.10	81.07	88.83	89.00
P-tuning (mtl) [8]*	95.04	90.35	91.10	88.05	90.65	82.72	90.16	89.72
TransPrompt [1]	94.61	90.80	91.25	87.93	91.62	85.90	91.47	90.57
TransPrompt v2	**95.53**	**91.15**	**91.95**	**88.49**	**92.17**	**86.30**	**91.53**	**91.10**

decreases. We guess that the semantics of sentence-pair tasks are highly different from single-sentence tasks, which improves the difficulty of knowledge transfer.

- Tasks from NLI and paraphrase can be enhanced from each other. As shown in Figure 5(b) and Figure 5(c), with the increase of training data, there is a slight improvement on the NLI and paraphrase tasks.

4.6 Learning with Full Datasets

Besides, we are also interested in how *TransPrompt v2* performs when learning with full training sets. We follow the base-scale experimental settings in [8], with the RoBERTa base model (with 109M parameters) as the underlying PLM. The results are presented in Table 7. On average, *TransPrompt v2* outperforms all baselines by around 1% to 5% regardless of whether the model is replaced in single-task or cross-task scenarios. This shows that our proposed paradigm can be of help in non-few-shot learning scenarios by learning from similar or distant tasks.

Another interesting finding is that when it comes to multi-task learning, the performance of LM-BFF [7] and P-tuning [8] drops, compared to the single-task setting. A most possible cause is that with a large amount of training data from other tasks, existing prompt-based approaches may capture non-transferable knowledge that is harmful to the target task. In contrast, the two-step paradigm of *TransPrompt v2* learns different types of knowledge at different steps (i.e., the universal knowledge in MMA, and the task-specific knowledge in TMS), hence producing better results. Overall, *TransPrompt v2*

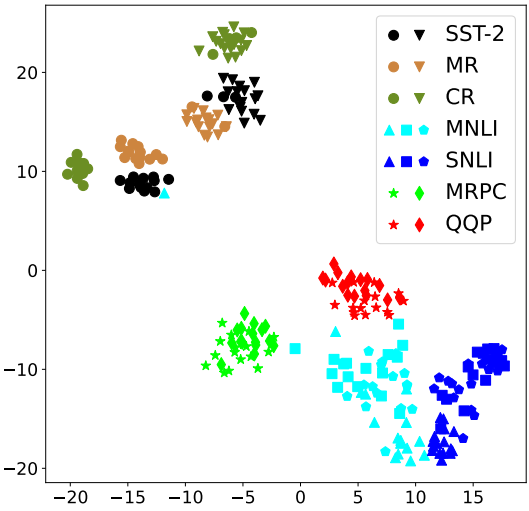


Fig. 6 Visualizations of sentence embeddings of few-shot scenarios over all datasets. Different colors represent different datasets. Different shapes represent different categories. (Best viewed in color.)

Table 8 Cases of review texts in SST-2, MR and CR with high and low cross-task prototype scores.

Score	Task	Review Text	Label
High	SST-2	There are many definitions of "time waster" but this movie must surely be one of them...	NEG
	MR	It is most remarkable not because of its epic scope, but because of the startling intimacy...	POS
	CR	5 stars all the way!	POS
Low	SST-2	It's a treat watching show, a British stage icon, melting under the heat of phocion's attentions.	POS
	MR	Humorous, artsy, and even cute, in an off-kilter, dark, vaguely disturbing way.	POS
	CR	However, the calls constantly drop in my area and I experience mega-static, to the point...	NEG

is competitive in standard fine-tuning scenarios with datasets from similar NLP tasks available.

4.7 Visualizations

Furthermore, we visualize one set of sentences from all development sets in the few-shot settings to gain more insight into the model performance. Specifically, we directly aggregate all few-shot training data from all datasets (i.e., 256 examples) to train the meta-learner, and then perform inference over corresponding few-shot development sets. For each sentence, we choose the representations at the position of `[MASK]` as the sentence embeddings. It can be seen from Figure 6 that our proposed technique plays an important role in sentence representations. Intuitively, we find that the tasks from sentiment analysis (which are single-sentence tasks) are more similar to each other, while NLI and paraphrase (which are sentence-pair tasks) have similar distributions. However, there exists a large difference between single-sentence and

sentence-pair tasks, which makes the transfer across distant tasks challenging. In addition, the results also demonstrate that the classification performance of sentiment analysis is better than others.

4.8 Case Studies

For a more initiative understanding of which instances are *more transferable across tasks*, in Table 8, several review texts from SST-2, MR, and CR with high and low prototype scores are presented. Although these texts come from different tasks, our algorithm is able to find texts that express general polarities instead of specific points. For instance, "time waster", "remarkable" and "5 stars" are strong indicators of polarities, which receive high scores generated. In contrast, review texts with low scores are overly specific and hence are less transferable across tasks. Hence, our meta-learner truly captures the transferable knowledge.

5 Limitations and Extensions

In this section, we discuss the limitations and extensions of *TransPrompt v2*, aiming to stimulate the research in this field.

5.1 Extending to Other NLP Tasks

The backbone of the proposed *TransPrompt v2* is primarily based on BERT [2] and RoBERTa [3], which can also be leveraged to solve other NLP tasks, such as sequence tagging [48], question answering [38]. This drives us to explore how *TransPrompt v2* can be learned across other format tasks. In addition, a lot of research works [38, 40] find that prompt-tuning can be used for text generation tasks with transformer encoders only. We also notice that, the PLMs with encoder-decoder structures including BART [33], T5 [34] and many others can also fine-tuned by prompt-based methods. Hence, *TransPrompt v2* is also capable of dealing these scenarios.

5.2 Incorporating Other Prior Knowledge

In this work, we find that task type descriptions make substantial contributions to distant cross-task learning, which can also be viewed as prior knowledge enhancement. Apart from the heuristic descriptions, we suggest that there are other structures or prior knowledge that can be used for distant cross-task learning, such as pre-trained knowledge base embeddings (KBE) from the knowledge graph, and common-sense knowledge from the encyclopedia (e.g., Wikipedia[7], WordNet [68], etc). For example, we can stack new initialized attentive fusion networks in prompt encoders to aggregate contextual representations with knowledge embeddings. We believe that the distant tasks with similar knowledge embeddings can be better transferable.

[7] https://www.wikipedia.org/.

6 Conclusion

In this paper, we present the *TransPrompt v2* framework for few-shot learning across similar and distant text classification tasks based on continuous prompt embeddings and task type description. Experimental results show that *TransPrompt v2* consistently outperforms strong baselines in both few-shot learning and standard fine-tuning settings. Additionally, we find that the meta-learner trained by *TransPrompt v2* can be adapted to previously unseen tasks easily. Future works include i) expanding our work to other multiple NLP tasks during cross-task learning. ii) exploring how *TransPrompt v2* is applied to other PLMs apart from BERT-style models. iii) devoting to the knowledge-enhanced prompt-based fine-tuning for distant cross-task learning.

Declarations

Ethical Approval and Consent to participate

Not applicable.

Human and Animal Ethics

Not applicable.

Consent for publication

Not applicable.

Availability of supporting data

All the data are mentioned in this manuscript.

Competing interests

The authors declare no competing interests.

Funding

This work was partially supported by Alibaba Group through the Alibaba Innovative Research Program and the National Natural Science Foundation of China under Grant Nos. U1911203, 61877018, 61672234, and 61672384.

Authors' contributions

Jianing Wang and Chengyu Wang wrote the main manuscript text. Jianing Wang implemented the model framework and the experiments. Cen Chen, Ming Gao, Jun Huang, and Aoying Zhou provided thoughtful suggestions for this manuscript.

Acknowledgments

The authors would like to acknowledge Chuanqi Tan and Songfang Huang from Alibaba DAMO Academy, Alibaba Group for the help of this research.

Authors' information

Jianing Wang, School of Science and Engineering (DaSE), East China Normal University (ECNU), lygwjn@gmail.com.
Chengyu Wang, Alibaba Group, chengyu.wcy@alibaba-inc.com.
Cen Chen, School of Data Science and Engineering (DaSE), East China Normal University (ECNU), cenchen@dase.ecnu.edu.cn.
Ming Gao, School of Data Science and Engineering (DaSE), East China Normal University (ECNU), mgao@dase.ecnu.edu.cn.
Jun Huang, Alibaba Group, huangjun.hj@alibaba-inc.com.
Aoying Zhou, School of Data Science and Engineering (DaSE), East China Normal University (ECNU), ayzhou@dase.ecnu.edu.cn.

References

[1] Wang, C., Wang, J., Qiu, M., Huang, J., Gao, M.: Transprompt: Towards an automatic transferable prompting framework for few-shot text classification. In: EMNLP, pp. 2792–2802 (2021)

[2] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL, pp. 4171–4186 (2019)

[3] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019)

[4] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A lite BERT for self-supervised learning of language representations. In: ICLR (2020)

[5] Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: NeurIPS, pp. 5754–5764 (2019)

[6] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.*: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

[7] Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) ACL, pp. 3816–3830 (2021)

[8] Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., Tang, J.: GPT understands, too. CoRR **abs/2103.10385** (2021)

[9] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: NeurIPS (2020)

[10] Schick, T., Schütze, H.: It's not just size that matters: Small language models are also few-shot learners. ACL, 2339–2352 (2021)

[11] Scao, T.L., Rush, A.M.: How many data points is a prompt worth? In: NAACL, pp. 2627–2636 (2021)

[12] Schick, T., Schütze, H.: Exploiting cloze-questions for few-shot text classification and natural language inference. In: EACL, pp. 255–269 (2021)

[13] Wang, C., Qiu, M., Huang, J., He, X.: Meta fine-tuning neural language models for multi-domain text mining. In: EMNLP, pp. 3094–3104 (2020)

[14] Wang, C., Pan, H., Qiu, M., Huang, J., Yang, F., Zhang, Y.: Meta distant transfer learning for pre-trained language models. In: EMNLP, pp. 9742–9752 (2021)

[15] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP, pp. 1631–1642 (2013)

[16] Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD, pp. 168–177 (2004)

[17] Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: ACL, pp. 115–124 (2005)

[18] Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference. In: NAACL, pp. 1112–1122 (2018)

[19] Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: EMNLP, pp. 632–642 (2015)

[20] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K.,

Zettlemoyer, L.: Deep contextualized word representations. In: NAACL, pp. 2227–2237 (2018)

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)

[22] Wang, W., Bi, B., Yan, M., Wu, C., Xia, J., Bao, Z., Peng, L., Si, L.: Structbert: Incorporating language structures into pre-training for deep language understanding. In: ICLR (2020)

[23] Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: Spanbert: Improving pre-training by representing and predicting spans. Trans. Assoc. Comput. Linguistics **8**, 64–77 (2020)

[24] He, P., Liu, X., Gao, J., Chen, W.: Deberta: decoding-enhanced bert with disentangled attention. In: ICLR (2021)

[25] Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., Wu, H.: ERNIE: enhanced representation through knowledge integration. CoRR **abs/1904.09223** (2019)

[26] Xiong, W., Du, J., Wang, W.Y., Stoyanov, V.: Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In: ICLR (2020)

[27] Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P.: K-BERT: enabling language representation with knowledge graph. In: AAAI, pp. 2901–2908 (2020)

[28] Sun, T., Shao, Y., Qiu, X., Guo, Q., Hu, Y., Huang, X., Zhang, Z.: Colake: Contextualized language and knowledge embedding. In: COLING, pp. 3660–3670 (2020)

[29] Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: enhanced language representation with informative entities. In: ACL, pp. 1441–1451 (2019)

[30] Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., Tang, J.: KEPLER: A unified model for knowledge embedding and pre-trained language representation. Trans. Assoc. Comput. Linguistics **9**, 176–194 (2021)

[31] Zhang, T., Cai, Z., Wang, C., Li, P., Li, Y., Qiu, M., Tang, C., He, X., Huang, J.: HORNET: enriching pre-trained language representations with heterogeneous knowledge sources. In: CIKM, pp. 2608–2617 (2021)

[32] Zhang, T., Wang, C., Hu, N., Qiu, M., Tang, C., He, X., Huang, J.:

DKPLM: decomposable knowledge-enhanced pre-trained language model for natural language understanding. CoRR **abs/2112.01047** (2021)

[33] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL, pp. 7871–7880 (2020)

[34] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140–114067 (2020)

[35] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. CoRR **abs/2107.13586** (2021)

[36] Shin, T., Razeghi, Y., IV, R.L.L., Wallace, E., Singh, S.: Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In: EMNLP, pp. 4222–4235 (2020)

[37] Han, X., Zhao, W., Ding, N., Liu, Z., Sun, M.: PTR: prompt tuning with rules for text classification. CoRR **abs/2105.11259** (2021)

[38] Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z., Tang, J.: P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. CoRR **abs/2110.07602** (2021)

[39] Zhong, Z., Friedman, D., Chen, D.: Factual probing is [MASK]: learning vs. learning to recall. In: NAACL, pp. 5017–5033 (2021)

[40] Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: ACL, pp. 4582–4597 (2021)

[41] Zhao, M., Schütze, H.: Discrete and soft prompting for multilingual models. In: EMNLP, pp. 8547–8555 (2021)

[42] Gu, Y., Han, X., Liu, Z., Huang, M.: PPT: pre-trained prompt tuning for few-shot learning. CoRR **abs/2109.04332** (2021)

[43] Hu, S., Ding, N., Wang, H., Liu, Z., Li, J., Sun, M.: Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. CoRR **abs/2108.02035** (2021)

[44] Sun, Y., Zheng, Y., Hao, C., Qiu, H.: NSP-BERT: A prompt-based zero-shot learner through an original pre-training task-next sentence prediction. CoRR **abs/2109.03564** (2021)

[45] Wang, S., Fang, H., Khabsa, M., Mao, H., Ma, H.: Entailment as few-shot learner. CoRR **abs/2104.14690** (2021)

[46] Li, C., Gao, F., Bu, J., Xu, L., Chen, X., Gu, Y., Shao, Z., Zheng, Q., Zhang, N., Wang, Y., Yu, Z.: Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. CoRR **abs/2109.08306** (2021)

[47] Chen, X., Zhang, N., Xie, X., Deng, S., Yao, Y., Tan, C., Huang, F., Si, L., Chen, H.: Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. CoRR **abs/2104.07650** (2021)

[48] Ma, R., Zhou, X., Gui, T., Tan, Y., Zhang, Q., Huang, X.: Template-free prompt tuning for few-shot NER. CoRR **abs/2109.13532** (2021)

[49] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A comprehensive survey on transfer learning. Proc. IEEE **109**(1), 43–76 (2021)

[50] Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)

[51] Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., Zhang, G.: Transfer learning using computational intelligence: A survey. Knowl. Based Syst. **80**, 14–23 (2015)

[52] Wang, B., Qiu, M., Wang, X., Li, Y., Gong, Y., Zeng, X., Huang, J., Zheng, B., Cai, D., Zhou, J.: A minimax game for instance based selective transfer learning. In: SIGKDD, pp. 34–43. ACM, ??? (2019)

[53] Liu, X., He, P., Chen, W., Gao, J.: Multi-task deep neural networks for natural language understanding. In: ACL, pp. 4487–4496 (2019)

[54] Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: CCL, pp. 194–206 (2019)

[55] Arase, Y., Tsujii, J.: Transfer fine-tuning of BERT with phrasal paraphrases. Comput. Speech Lang. **66** (2021)

[56] Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: A survey on few-shot learning. ACM Comput. Surv. **53**(3), 63–16334 (2020)

[57] Huisman, M., van Rijn, J.N., Plaat, A.: A survey of deep meta-learning. Artif. Intell. Rev. **54**(6), 4483–4541 (2021)

[58] Geng, R., Li, B., Li, Y., Sun, J., Zhu, X.: Dynamic memory induction networks for few-shot text classification. In: ACL, pp. 1087–1094 (2020)

[59] Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., Zhou, J.: Fewrel 2.0: Towards more challenging few-shot relation classification. In: EMNLP, pp. 6249–6254 (2019)

[60] Hua, Y., Li, Y., Haffari, G., Qi, G., Wu, T.: Few-shot complex knowledge base question answering via meta reinforcement learning. In: EMNLP, pp. 5827–5837 (2020)

[61] Sheng, J., Guo, S., Chen, Z., Yue, J., Wang, L., Liu, T., Xu, H.: Adaptive attentional network for few-shot knowledge graph completion. In: EMNLP, pp. 1681–1691 (2020)

[62] Pan, H., Wang, C., Qiu, M., Zhang, Y., Li, Y., Huang, J.: Meta-kd: A meta knowledge distillation framework for language model compression across domains. In: ACL/IJCNLP, pp. 3026–3036 (2021)

[63] Wang, C., Pan, H., Liu, Y., Chen, K., Qiu, M., Zhou, W., Huang, J., Chen, H., Lin, W., Cai, D.: Mell: Large-scale extensible user intent classification for dialogue systems with meta lifelong learning. In: KDD, pp. 3649–3659 (2021)

[64] Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: NIPS, pp. 4077–4087 (2017)

[65] Jamal, M.A., Qi, G.: Task agnostic meta-learning for few-shot learning. In: CVPR, pp. 11719–11727 (2019)

[66] Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: IWP@IJCNLP (2005)

[67] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)

[68] Piasecki, M., Szpakowicz, S., Fellbaum, C., Pedersen, B.S.: Introduction to the special issue: On wordnets and relations. Lang. Resour. Evaluation **47**(3), 757–767 (2013)

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- TransPromptv2.zip