Differentiable Prompt Makes Pre-trained Language Models Better Few-shot Learners

Ningyu Zhang[1,2*] **Luoqiu Li**[1,2*] **Xiang Chen**[1,2*] **Shumin Deng**[1,2*] **Zhen Bi**[1,2]
Chuanqi Tan[3] **Fei Huang**[3] **Huajun Chen**[1,2†]
[1]Zhejiang University & AZFT Joint Lab for Knowledge Engine
[2]Hangzhou Innovation Center, Zhejiang University
[3]Alibaba Group
{zhangningyu,3160102409,xiang_chen,231sm,bizhen_zju}@zju.edu.cn,
{chuanqi.tcq,f.huang}@alibaba-inc.com

Abstract

Large-scale pre-trained language models have contributed significantly to natural language processing by demonstrating remarkable abilities as few-shot learners. However, their effectiveness depends mainly on scaling the model parameters and prompt design, hindering their implementation in most real-world applications. This study proposes a novel pluggable, extensible, and efficient approach named DifferentiAble pRompT (DART), which can convert small language models into better few-shot learners without any prompt engineering. The main principle behind this approach involves reformulating potential natural language processing tasks into the task of a pre-trained language model and differentially optimizing the prompt template as well as the target label with backpropagation. Furthermore, the proposed approach can be: (i) Plugged to any pre-trained language models; (ii) Extended to widespread classification tasks. A comprehensive evaluation of standard NLP tasks demonstrates that the proposed approach achieves a better few-shot performance.

1 Introduction

The pre-train—fine-tune paradigm has become the de facto standard for natural language processing (NLP), and has achieved excellent results in several benchmarks [10, 27, 22, 13, 3]. The success of these pioneers seems to suggest that large-scale pre-trained models are always nothing short of a panacea for boosting machine intelligence. However, supervised fine-tuning is still prone to labeled data in practice and faces unignorable challenges owing to the variations of domains, language, and tasks. These drawbacks lead to the research of an important technique, *few-shot learning*, which can significantly improve the learning capabilities of machine intelligence and practical adaptive applications by accessing only a small number of labeled examples.

The GPT-3 model, introduced by [6], exhibits impressive few-shot learning capabilities. Given a natural language prompt and 16 labeled samples as demonstrations in the contextual input, GPT-3 achieves 80% of the SOTA results. However, GPT-3 is a fully dense transformer model with 175B parameters, which makes it challenging to deploy in most real-world applications.

Recently, an emerging fine-tuning methodology has arisen to equip smaller language models (LMs) with few-shot capabilities: adapting the pre-trained LM directly as a predictor through completion of a cloze task [39, 38, 14, 26], which treats the downstream task as a (masked) language modeling

*Equal contribution and shared co-first authorship.
†Corresponding author.

Preprint. Under review.

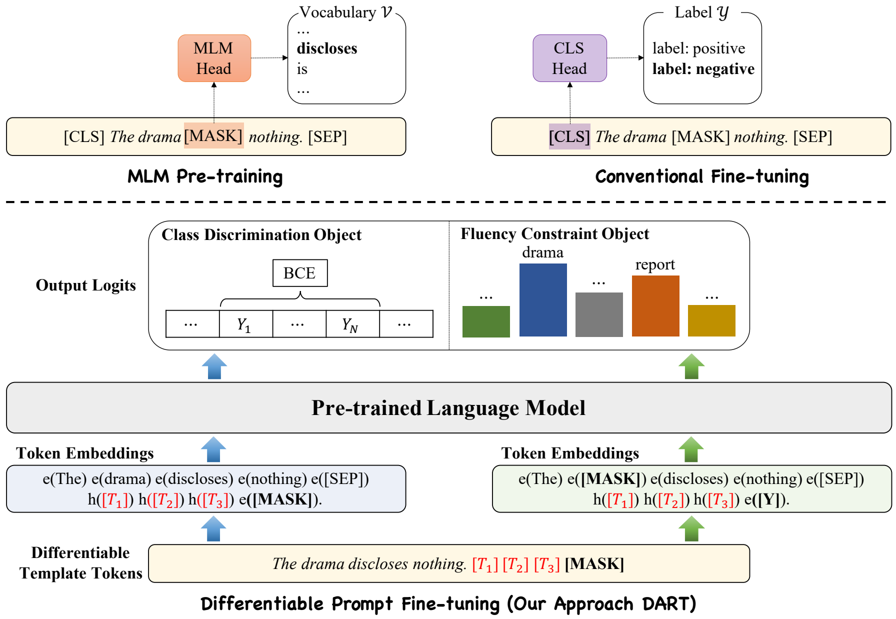


Figure 1: The architecture of **D**ifferenti**A**ble p**R**omp**T** (**DART**) model comparing with *MLM pre-training* and *conventional fine-tuning*, where T_i and Y_i are unused or special tokens in the vocabulary.

problem. These prompts can be used in fine-tuning to provide the classifier with additional task information, especially in the low-data regime. Notably, Scao et al. [37] observe that prompting can often compensate for hundreds of data points on average across multiple classification tasks. However, determining the appropriate prompts requires domain expertise, and handcrafting a high-performing prompt often requires impractically large validation sets [32]. Recent studies [28, 51] have reported that the manual prompt format can be sub-optimal, which would result in the accuracy varying from random guess performance to near the state-of-the-art. Therefore, previous approaches have attempted to search for discrete prompt tokens automatically. However, it is non-trivial for widespread classification tasks to obtain an optimized prompt template and target label token. For example, specific classification tasks such as relation extraction with the label of $alternate\_name$ and $country\_of\_birth$ cannot specify a single label token in the vocabulary.

In this paper, we propose a novel **D**ifferenti**A**ble p**R**omp**T** (**DART**) fine-tuning approach, which is model-agnostic, parameter-efficient, and free of prompt engineering. As illustrated in Figure 1, the key idea is to leverage a few parameters (unused tokens) in the language model, which serve as the template and label tokens, and to optimize them in the continuous space using backpropagation. Subsequently, we introduce differentiable prompt learning to obtain optimized prompt templates as well as labels. Since fine-tuning with limited samples can be affected by instability [12, 50], we propose a two-stage optimization algorithm of first learning templates and labels then the overall parameters. We further introduce an auxiliary fluency constraint object to ensure the association among the prompt embeddings.

We conduct extensive experiments on 15 NLP datasets. With only a few training samples across all the tasks, our approach (DART) can obtain a better performance. Notably, absolute performance improvement of up to **23.28%**, over the conventional fine-tuning, is obtained on average in the setting of $K = 8$ (and **1.55%** for fully supervised settings) on relation extraction datasets with complex label semantics. Our approach can be applied to real-world classification tasks without the high cost of collecting and annotating a large amount of data. The main contributions of this study are as follows:

- We propose a new simple framework for few-shot learning, which is pluggable, extensible, and efficient without prompt engineering. To the best of our knowledge, optimizing label tokens in continuous space is also a new branch of research that has not been explored in language model prompting.

- A systematic evaluation of 15 NLP tasks shows that the simple-yet-effective method contributes towards improvements across all these tasks. Remarkably, given only 8 labeled samples per class, our proposed approach can achieve 90% performance of the SOTA results (full dataset).

2 Related Work

Language Model Prompting. The language model prompting has emerged with the introduction of GPT-3 [6], which demonstrates excellent few-shot performance [25]. However, GPT-3 is not designed for fine-tuning; it mainly relies on the handcraft prompt (in-context learning [24, 51, 11, 29]). Thus, recent studies [34, 15, 7] conducted in this field have been focused on automatically searching the prompts. Schick et al. [39, 38] propose the PET, which reformulates the NLP tasks as cloze-style questions and performs gradient-based fine-tuning. Tam et al. [42] improve the PET with a denser supervision object during fine-tuning. Shin et al. [40] propose the AUTOPROMPT to create prompts for a diverse set of tasks based on a gradient-guided search. Han et al. [17] propose an approach called PTR which leverages logic rules to construct prompts with sub-prompts for many-class text classification. Wang et al. [44] reformulate potential NLP task into an entailment one, and then fine-tune the model with few-shot samples. Hu et al. [19] propose an approach to incorporate external knowledge graph into the verbalizer with calibration. Additionally, Gao et al. [14] present LM-BFF—better few-shot fine-tuning of language models, which leverages T5 [35] to generate templates and search label tokens in the vocabulary. However, the utilization of the generative model and the label search with validation is computation-intensive. Moreover, the prompt search over discrete space is sub-optimal due to the continuous nature of neural networks.

To overcome these limitations, Liu et al. [26] propose P-tuning, which employs trainable continuous prompt embeddings learned by an LSTM. Zhong et al. [52] propose an effective continuous method called OPTIPROMPT to optimize prompts for factual probing. Li et al. [26] propose prefix-tuning, which keeps language model parameters frozen but optimizes a small continuous task-specific vector for natural language generation tasks. Lester et al. [21] propose a mechanism for learning "soft prompts" to condition frozen language models to perform downstream tasks. However, these approaches still have to optimize the external parameters (e.g., LSTM in P-tuning) and are prone to complex label space.

Conversely, this study aims to develop a novel few-shot learning framework based on pre-trained language models which do not require prompt engineering (including templates and labels) and external parameter optimization. Furthermore, the proposed approach only leverages the noninvasive modification of the model, which can be plugged into any pre-trained language model and extended to the widespread classification task.

Few-shot Learning. Few-shot learning can significantly improve the learning capabilities for machine intelligence and practical adaptive applications by accessing only a small number of labeled examples [49]. The proposed approach corresponds to the other few-shot NLP methods, including: (1) Meta-learning [48, 4, 2, 9, 8, 47], in which the quantities of the auxiliary tasks are optimized. (2) Intermediate training [33, 46], which supplements the pre-trained LMs with further training on the data-rich supervised tasks. (3) Semi-supervised learning [30, 45], which leverages unlabeled samples. The proposed approach focuses on a more realistic few-shot setting (the number of labeled instances per class can be any variable).

3 Background

3.1 Language Model Prompting

Let $X_{\text{in}} = \{x_1, x_2, ..., x_L\}$ be a sentence, where x_i is the i^{th} token in the input sentence and L is the number of tokens. Specifically, X_{in} is converted to a fixed token sequence $\tilde{X}_{\text{in}}$ and then mapped to a sequence of hidden vectors $\{\mathbf{h}_k \in \mathbb{R}^d\}$. Given the input sequence, $\tilde{X}_{\text{in}} = \texttt{[CLS]}X_{\text{in}}\texttt{[SEP]}$, the conventional fine-tuning approaches leverage a generic head layer over [CLS] embeddings (e.g., an MLP layer) to predict an output class. For the prompt-based method, a task-specific pattern string (template $\mathcal{T}$) is designed to coax the model into producing a textual output corresponding to a given class (label token $\mathcal{M}(Y)$)—we refer to these two things together as a prompt. Specifically, X_{prompt} containing one `[MASK]` token is directly tasked with the MLM input as:

$$X_{\text{prompt}} = \texttt{[CLS]} X_{\text{in}} \ \texttt{[SEP]} \mathcal{T} \ \texttt{[SEP]} \tag{1}$$

When the prompt is fed into the MLM, the model can obtain the probability distribution $p(\texttt{[MASK]}|(X_{\text{prompt}})$ of the candidate class, $y \in Y$ as:

$$p(y|X_{\text{prompt}}) = \sum_{w \in \mathcal{V}_y} p(\texttt{[MASK]} = w|X_{\text{prompt}}) \tag{2}$$

where w represents the w^{th} label token of class y. To further understand the mechanism of language model prompting, we theoretically analyze the underlying intuitions.

4 Our Approach

4.1 Motivation

It can be observed from the previous empirical findings [14, 37] that an optimal prompt is necessary for the improvement of the pre-trained language models for the few-shot learners. Since templates with discrete tokens may be sub-optimal and are insufficient to represent a specific class[3], this study proposes **D**ifferenti**A**ble p**R**omp**T**, referred to as **DART**, which can reduce the requirement of prompt engineering in order to improve the applicability of the proposed method in various domains.

4.2 Differentiable Template Optimization

Since the language tokens are discrete variables, finding the optimal prompts with token searching is non-trivial and may easily fall into the local minima. To overcome these limitations, we utilize pseudo tokens to construct templates and then optimize them with backpropagation. Specifically, given the template, $\mathcal{T} = \{[\mathrm{T}_{0:i}], \texttt{[MASK]}, [\mathrm{T}_{i+1:j}]\}$, which varies from the traditional discrete prompts, satisfying $[\mathrm{T}_i] \in \mathcal{V}$ and map $\mathcal{T}$ into:

$$\{\mathbf{w}([\mathrm{T}_{0:i}]), \mathbf{w}(\texttt{[MASK]}), \mathbf{w}([\mathrm{T}_{i+1:m}])\} \tag{3}$$

DART considers $[\mathrm{T}_i]$ as pseudo tokens and maps the template as follows:

$$\{h_0, ..., h_i, \mathbf{w}(\texttt{[MASK]}), h_{i+1}, ..., h_m\} \tag{4}$$

where $h_i (0 \leq i \leq j)$ are trainable parameters. Differentiable template optimization can obtain expressive templates beyond the original vocabulary $\mathcal{V}$. Lastly, the templates, h_i, are differentially optimized by:

$$\hat{h}_{0:m} = \arg\min_h \mathcal{L}(X_{\text{prompt}}, y) \tag{5}$$

Note that the values of the prompt embeddings, h_i, must be co-dependent with each other rather than independent. Unlike P-tuning [26], which utilizes a bidirectional LSTM, DART leverages an auxiliary fluency constraint objective without any external parameters to associate the prompt embeddings with each other, thus stimulating the model to focus on context representation learning.

4.3 Differentiable Label Optimization

Prompt-based fine-tuning requires filling in one word, and the masked word prediction is mapped to a verbalizer, which produces a class (i.e., "Yes": True. "No": False). For each class $c \in Y$, the previous approaches such as LM-BFF [14] estimate the conditional likelihood of the initial $\mathcal{L}$ on a pruned set $\mathcal{V}^c \subset \mathcal{V}$ of the top k vocabulary words.

[3]It is non-trivial to evaluate all options of templates and label tokens.

However, the brute-forcing label searching: (1) is computationally intensive and tedious because the $\mathcal{D}_{\text{dev}}$ is generally very large, requiring multiple rounds of evaluation. (2) has poor scalability with an increase in the class numbers (many classification datasets have more than 100 classes), the number of searches may be k^C (C represents the total number of classes), which is exponential and thus intractable. Additionally, the labels of classes contain rich, complex semantic knowledge, and one discrete token may be insufficient to represent this information.

Specifically, with the labels, $Y = \{Y_1, Y_2, .., Y_m\}$, different from the previous approach which converts the class type Y_i into a variable number of label tokens $\{...,v_1,..,v_k,...\}$, DART maps the Y_j to a continuous vocabulary space as follows:

$$\mathcal{M}(Y_j) = \{h_{m+j}\}, \tag{6}$$

, where m is the number of trainable embedding in template. To avoid optimizing any external parameters, $\{h_1, ..., h_m, .., h_{m+n}\}$ is replaced with unused tokens (e.g., [unused1] or special tokens in vocabulary) in $\mathcal{V}$ to generate $\mathcal{V}'$, as shown in Figure 1.

4.4 Training Objectives

Since the pseudo tokens in the prompt template must be co-dependent with each other, we introduce an auxiliary fluency constraint training without optimizing any other parameters inspired by [26, 42]. Overall, there are two objectives: the class discrimination objective $\mathcal{L}_C$ and the fluency constraint objective $\mathcal{L}_F$.

Class Discrimination Object The class discrimination objective is the main objective which aims to classify the sentences. As shown in Figure 1, given $(X_{\text{in}}, \mathcal{T})$, we can generate X_{prompt} as:

$$\mathcal{L}_C = \text{CE}(p(y|X_{\text{prompt}})). \tag{7}$$

where CE is the cross-entropy loss function, $\mathcal{L}_C$ represents the class discrimination loss.

Fluency Constraint Object To ensure the association among the template tokens and to maintain the ability of language understanding inherited from the PLMs, we leverage a fluency constraint object with the MLM. As shown in Figure 1, one token in the input sentence is randomly masked and the masked language prediction is conducted. x and x' are the original and masked sequences, respectively. Let x^m be the target token that has been masked out in x', and $P(x^m|x', y)$ is maximized as follows[4]:

$$q(x^m|x', y) = \frac{\exp([\![L(x', y)]\!]_{x^m})}{\sum\limits_{v' \in \mathcal{V}'} \exp([\![L(x', y)]\!]_{v'})} \tag{8}$$

$$\mathcal{L}_F = \sum_{m \in M} \text{BCE}(q(x^m|x', y)). \tag{9}$$

By optimizing $\mathcal{L}_F$, the language model can obtain a better contextual representation with a rich association among the template tokens. We have the following training object:

$$\mathcal{L} = \mathcal{L}_C + \lambda \mathcal{L}_F, \tag{10}$$

where λ is the hyper-parameter. Lastly, we introduce the overall optimization procedure of DART. To mitigate the instability of the few-shot fine-tuning, we introduce a two-stage optimization algorithm in Algorithm 1. At optimization step (lines 2–6), the $\{h_0, ..., h_m, .., h_{m+n}\}$ described in §4.2 and §4.3 is first optimized to obtain the optimal prompts (We implement the procedure via stopping the gradient of other parameters), and is then all parameters are optimized in the step (lines 7–11).

[4] We use the golden label y rather than the [MASK] in the input of the fluency constraint object.

Algorithm 1 Differentiable Prompt Fine-tuning Algorithm with Two-stage Optimization

Require: $\mathcal{J}(\theta)$: stochastic objective function with parameters θ; α,β: learning rate; $h \in \theta$: parameters of the templates and label tokens;

initialize $t \leftarrow 0$;
while h_t not converged **do** ▷ Template and label optimization with learning rate α
  $t \leftarrow t+1$;
  $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$;
  $h_t \leftarrow AdamW(g_t, \alpha)$;
end while
while θ_t not converged **do** ▷ All parameter optimization with learning rate β
  $t \leftarrow t+1$;
  $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$;
  $\theta_t \leftarrow AdamW(g_t, \beta)$;
end while

4.5 Comparison to Previous Prompt-tuning Approaches

Since prompt learning has become a new paradigm or a way for human-PLMs communication, it appeals to many researchers. Due to the fast development of prompt learning, some similar ideas (learned embeddings) may be introduced by different research teams. However, we list the major difference between our model and other approaches as shown in Table 1:

To conclude, our approach is quite simple and requires no external parameters (different from WARP/Prefix-Tuning/P-tuning/ADAPET). Moreover, our approach unifies the optimization of template and answer.

Model	External Parameter	External Architecture	Template	Answer
Prefix-Tuning [23]	yes	no	continuous	no
WARP [15]	yes	no	continuous	continuous
P-tuning [26]	yes	LSTM	continuous	no
ADAPET [42]	yes	no	discrete	discrete
DART (Ours)	no	no	continuous	continuous

Table 1: The difference between DART and previous prompt-tuning approaches.

5 Experiments

In this section, we detail the comprehensive experimental results conducted on classification tasks. The promising results demonstrate that our proposed DART substantially outperforms the conventional fine-tuning method, thus, making pre-trained language models better few-shot learners.

5.1 Dataset Statistics

We conduct a comprehensive study across 15 NLP tasks, which covers sentiment analysis, natural language inference, paraphrases, sentence similarity, relation extraction, and event extraction (We only report event argument extraction performance). The evaluation consisted of 10 popular sentence classification datasets (SST-2, MR, CR, Subj, TREC, MNLI, SNLI, QNLI, MRPC, QQP).To further evaluate the effectiveness of the proposed approach with complex label space, we conduct experiments on the relation extraction and event extraction datasets, including SemEval-2010 Task 8 [18], TACRED-Revisit [1], Wiki80[5] [16], ChemProt [20], and ACE-2005[6].

[5] `https://github.com/thunlp/OpenNRE/`

[6] `https://catalog.ldc.upenn.edu/LDC2006T06`

Model	**SST-2** (acc)	**MR** (acc)	**CR** (acc)	**Subj** (acc)	**TREC** (acc)
Majority†	50.9	50.0	50.0	50.0	18.8
Prompt-based zero-shot‡	83.6	80.8	79.5	51.4	32.0
"GPT-3" in-context learning	84.8 (1.3)	80.5 (1.7)	87.4 (0.8)	53.6 (1.0)	26.2 (2.4)
Fine-tuning	81.4 (3.8)	76.9 (5.9)	75.8 (3.2)	90.8 (1.8)	88.8 (2.1)
LM-BFF	92.3 (1.0)	85.5 (2.8)	89.0 (1.4)	91.2 (1.1)	88.2 (2.0)
P-Tuning	92.2 (0.4)	86.7 (1.2)	91.8 (1.1)	90.3 (2.2)	86.3 (4.5)
DART	**93.5 (0.5)**	**88.2 (1.0)**	**91.8 (0.5)**	90.7 (1.4)	87.1(3.8)
Fine-tuning (full)†	*95.0*	*90.8*	*89.4*	*97.0*	*97.4*

Model	**MNLI** (acc)	**SNLI** (acc)	**QNLI** (acc)	**MRPC** (F1)	**QQP** (F1)
Majority†	32.7	33.8	49.5	81.2	0.0
Prompt-based zero-shot‡	50.8	49.5	50.8	61.9	49.7
"GPT-3" in-context learning	52.0 (0.7)	47.1 (0.6)	53.8 (0.4)	45.7 (6.0)	36.1 (5.2)
Fine-tuning	45.8 (6.4)	48.4 (4.8)	60.2 (6.5)	76.6 (2.5)	60.7 (4.3)
LM-BFF	68.3 (2.5)	77.1 (2.1)	68.3 (7.4)	76.2 (2.3)	67.0 (3.0)
P-Tuning	61.5 (2.1)	72.3 (3.0)	64.3 (2.8)	74.5 (7.6)	65.6 (3.0)
DART	67.5 (2.6)	75.8 (1.6)	66.7 (3.7)	**78.3 (4.5)**	**67.8 (3.2)**
Fine-tuning (full)†	*89.8*	*92.6*	*93.3*	*91.4*	*81.7*

Table 2: Our main results with RoBERTa-large. †: the full training set is used. ‡: no training examples are used. Otherwise, we use $K = 16$ (# examples per class). We report mean (and standard deviation) performance over 5 different splits. Majority: majority class "GPT-3" in-context learning: using the in-context learning proposed in with RoBERTa-large (no parameter updates); LM-BFF: we report the performance in [14]. full: fine-tuning using full training set.

5.2 Settings

The proposed model is implemented using Pytorch [31] and Our experiments are conducted with the same setting following LM-BFF [14], which measures the average performance with a fixed set of seeds, $\mathcal{S}_{\text{seed}}$, across five different sampled $\mathcal{D}_{\text{train}}$ for each task. We utilize a grid search over multiple hyperparameters and select the best result as measured on $\mathcal{D}_{\text{dev}}$ for each set $\{\mathcal{D}^s_{\text{train}}, \mathcal{D}_{\text{dev}}\}, s \in \mathcal{S}_{\text{seed}}$. We employ AdamW as the optimizer. We conduct experiments with a RoBERTa-large [27] on classification tasks for a fair comparison with LM-BFF. We leverage an uncased BERT-large [10] for relation extraction datasets, except that we use SCIBERT [5] for the ChemProt dataset. We follow [41] and use special entity markers uniformly to highlight the entity mentions for relation extraction.

5.3 Main Results

As shown in Table 2, we observe that our approach obtains better performance than conventional fine-tuning and achieves comparable results with LM-BFF. Note that DART does not need any prompt engineering or external model (e.g., T5 in LM-BFF) to generate templates that are readily easy to adapt to other datasets. DART can obtain **11.3%** improvement with only 16 training samples per class on the MR dataset, comparable with LM-BFF, which leverages T5 to generate appropriate prompts. These results indicate that DART can better stimulate potential ability and makes the pre-trained language model a better few-shot learner. We also notice that DART yields better performance than P-tuning, which indicates that label optimization is beneficial.

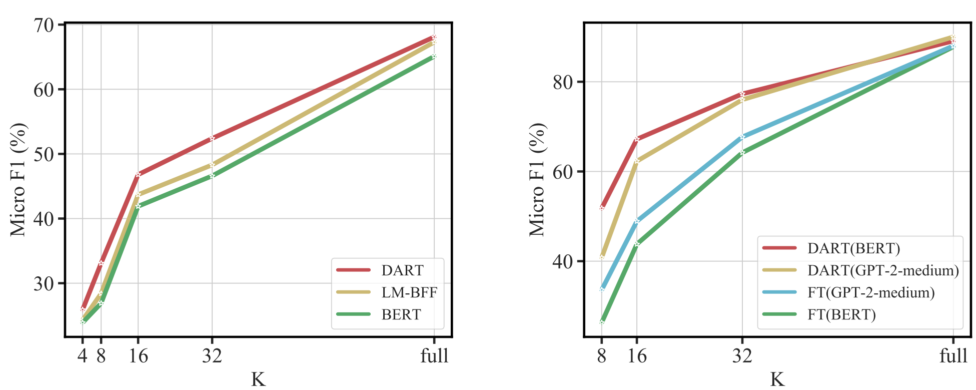
For the classification tasks with the complex label space, as shown in Table 3 and Figure 2(a), we observe that DART outperforms the conventional fine-tuning approach as well as LM-BFF with a large margin on relation extraction and event extraction datasets in both the few-shot and fully supervised settings. The proposed approach achieves an improvement of **2.8%** of the absolute performance on the TACRED-Revisit dataset with full supervision and yields **18.4%** gains with only 8 training samples per class. These findings also indicate that more relevant templates and labels can be determined without expert intervention, making it possible to generalize the proposed approach to other domains. Furthermore, we notice that the improvement decays slowly when K becomes larger (i.e., from 8 to 32). Our approach is a simple yet effective fine-tuning paradigm that does not require prompt engineering within the complex label space, thus, making it possible to be an appropriate plug-in for some SOTA models.

Dataset	Model	$K=8$	$K=16$	$K=32$	$Full$
SemEval	Fine-tuning	26.3	43.8	64.2	87.8
	LM-BFF	43.2	62.0	72.9	88.0
	DART	**51.8** (+25.5)	**67.2** (+23.4)	**77.3** (+13.1)	**89.1** (+1.3)
TACRED-Revisit	Fine-tuning	7.4	15.5	25.8	75.0
	LM-BFF	21.0	23.7	27.1	76.4
	DART	**25.8** (+18.4)	**30.1** (+14.6)	**31.8** (+6.0)	**77.8** (+2.8)
WiKi80	Fine-tuning	46.3	60.3	70.0	87.5
	LM-BFF	66.5	73.5	78.1	86.2
	DART	**68.5** (+22.2)	**75.2** (+14.9)	**79.4** (+9.4)	**88.1** (+0.6)
ChemProt	Fine-tuning	30.2	41.5	52.5	79.5
	LM-BFF	55.0	56.1	60.0	79.1
	DART	**57.2** (+27.0)	**60.8** (+19.3)	**63.1** (+10.6)	**81.0** (+1.5)

Table 3: Results on RE dataset WiKi80 (accuracy), while other datasets (micro F_1). We use $K = 8, 16, 32$ (# examples per class). $Full$ represents the full training set is used.

Method	K=8	K=16	K=32	Full
Conventional FT	26.3	43.8	64.2	87.8
DART	**51.8**	**67.2**	**77.3**	**89.1**
-fluency constraint object	50.3	66.1	76.0	88.2
-differentiable template	49.8	66.3	76.2	88.4
-differentiable label	47.5	62.5	73.7	87.8

Table 4: Ablation of DART with different components on SemEval. (FT= Fine tuning)



(a) Event extraction results on ACE-2005. (b) BERT-large & GPT-2-medium results on SemEval.

Figure 2: (a) Few-shot results using the ACE-2005. We used K = 4, 8, 16, and 32 (# examples per class) with BERT. (FT= Fine-tuning) (b) BERT-large vs. GPT-2-medium results for the SemEval. Moreover, for lower K, our method consistently outperforms conventional fine-tuning.

5.4 Ablation Study

We conduct an ablation study to validate the effectiveness of the components in the proposed approach. We observe that DART exhibits a performance decay in the absence of any one of the modules, i.e., fluency constraint object, differentiable template, or differentiable label, demonstrating that all the modules are advantageous. Furthermore, we notice that differentiable label optimization is more sensitive to performance and is highly beneficial for DART, especially for low-resource settings. Since the proposed approach is the first approach that utilizes the differentiable label optimization, these findings illustrate that a suitable label token is important.

5.5 Analysis and Discussion

Can DART Applied to Other Pre-trained LMs?

To evaluate whether the proposed approach can be applied to other LMs, we conduct experiments using GPT-2-medium. From Figure 2(b), we observe that DART with GPT-2-medium yields better performance than the conventional fine-tuning approach. Furthermore, we notice that DART with GPT-2-medium can achieve performance on par with BERT-large, as observed by [26], indicating that the potential of GPT-style architectures for natural language understanding has been underestimated.

What Exactly Optimized Prompt is?

Since prompt templates and label tokens in the proposed approach are mapped as $\{h_1, ..., h_m, .., h_{m+n}\}$, we further analyze what exactly optimized label learned. We conduct a nearest-neighbor vocabulary embedding search to project the Top-3 optimized pseudo-label tokens in $\mathcal{V}$ to a readable natural language.We use t-SNE [43] with normalization to visualize labels on Wiki80 dataset. For example, "*military_branch*" refers to as red $\star$ in Figure 3 represents the relation type, which is learned by optimizing the pseudo label in the continuous space, and the "*volunteered*", "*corporal*" and "*buddies*", refers to as $\bullet$ are the tokens closest to the label. This finding indicates that optimized label embeddings can present better semantic representation ability.

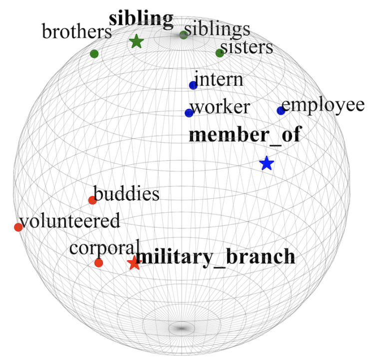


Figure 3: A 3D visualization of several label representations learned in DART on WiKi80 dataset with t-SNE and normalization.

DART v.s. Conventional Fine-tuning

The ability of the proposed approach to perform few-shot learning can be attributed to the label and being a true language understanding task, that once the model is capable of performing it correctly, it can easily apply this knowledge to other tasks that are framed as such. Superficially, (i) DART does not optimize any new parameters; however, conventional fine-tuning should learn an explicit classifier head over [CLS] embeddings, which may fail in the low-data regime. (ii) DART has the same task setting as large-scale language model pre-training and has a small theoretical upper bound for downstream classification tasks [36].

Limitations

Our work may fail when the distribution of the task corpus varies from that of the pre-training corpus. For example, a general pre-trained language model may be fine-tuned with more training instances in a specific domain (e.g., medical domain). This issue can be addressed by intermediate training [33, 46, 51], and will be analyzed in the future work. Besides, our work also shows an instability associated with hyper-parameters which is also observed by [12, 50, 32] as volatility of few-shot learning in NLP. Overall, however, we believe our work will inspire future work to few-shot settings with more practical applications to low-data settings, e.g., that involve low-resource languages or expert annotation.

6 Conclusion and Future Work

This paper presents DART, a simple-yet-effective fine-tuning approach that improves the fast-shot learning pre-trained language model. The proposed approach can produce satisfactory improvements in the few-shot scenarios when compared to the conventional fine-tuning approaches. The proposed method is also pluggable for other language models and can be extended to other tasks, such as intent detection. Intuitively, the results obtained in this study can be used to stimulate two future research directions in the few-shot learning for NLP: (i) Extending the proposed approach to a semi-supervised setting to further leverage unlabeled data; (ii) Extending the proposed approach to few-shot lifelong learning, whereas prompts must be optimized with adaptive tasks.

Broader Impact

The pre-train-fine-tune approach has become the standard for natural language processing (NLP). However, supervised fine-tuning is still practically affected by labeled data. This study proposes a novel pluggable, extensible, and efficient approach named DifferntiAble pRompT (DART), which can convert small language models into better few-shot learners without any prompt engineering. We believe that our study makes a significant contribution to the literature because determining the appropriate prompts requires domain expertise, and handcrafting a high-performing prompt often requires impractically large validation sets, and these issues have been overcome with the use of the proposed method, which is model-agnostic, parameter-efficient, and independent of prompt engineering. We experimentally verified our proposed approach on 13 standard NLP tasks, and it was seen to outperform several standard NLP platforms.

References

[1] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1558–1569. Association for Computational Linguistics, 2020.

[2] Trapit Bansal, Rishikesh Jha, and Andrew McCallum. Learning to few-shot learn across diverse natural language classification tasks. In Donia Scott, Núria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5108–5123. International Committee on Computational Linguistics, 2020.

[3] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 642–652. PMLR, 2020.

[4] Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. Few-shot text classification with distributional signatures. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[5] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3613–3618. Association for Computational Linguistics, 2019.

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[7] Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. *arXiv preprint arXiv:2104.07650*, 2021.

[8] Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang, editors, *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 151–159. ACM, 2020.

[9] Shumin Deng, Ningyu Zhang, Zhanlin Sun, Jiaoyan Chen, and Huajun Chen. When low resource NLP meets unsupervised language model: Meta-pretraining then meta-learning for few-shot text classification (student abstract). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13773–13774. AAAI Press, 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[11] Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*, 2021.

[12] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305, 2020.

[13] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054, 2019.

[14] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *CoRR*, abs/2012.15723, 2020.

[15] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. WARP: word-level adversarial reprogramming. *CoRR*, abs/2101.00121, 2021.

[16] Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. Opennre: An open and extensible toolkit for neural relation extraction. In Sebastian Padó and Ruihong Huang, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019 - System Demonstrations*, pages 169–174. Association for Computational Linguistics, 2019.

[17] Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259, 2021.

[18] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Katrin Erk and Carlo Strapparava, editors, *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pages 33–38. The Association for Computer Linguistics, 2010.

[19] Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *CoRR*, abs/2108.02035, 2021.

[20] Jens Kringelum, Sonny Kim Kjærulff, Søren Brunak, Ole Lund, Tudor I. Oprea, and Olivier Taboureau. Chemprot-3.0: a global chemical biology diseases mapping. *Database J. Biol. Databases Curation*, 2016, 2016.

[21] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691, 2021.

[22] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020.

[23] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *CoRR*, abs/2101.00190, 2021.

[24] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? *CoRR*, abs/2101.06804, 2021.

[25] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586, 2021.

[26] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT understands, too. *CoRR*, abs/2103.10385, 2021.

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[28] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *CoRR*, abs/2104.08786, 2021.

[29] Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Noisy channel language model prompting for few-shot text classification. *CoRR*, abs/2108.04106, 2021.

[30] Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019.

[32] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. *arXiv preprint arXiv:2105.11447*, 2021.

[33] Jason Phang, Thibault Févry, and Samuel R. Bowman. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *CoRR*, abs/1811.01088, 2018.

[34] Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. *CoRR*, abs/2104.06599, 2021.

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[36] Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. A mathematical exploration of why language models help solve downstream tasks. In *International Conference on Learning Representations*, 2021.

[37] Teven Le Scao and Alexander M. Rush. How many data points is a prompt worth? *CoRR*, abs/2103.08493, 2021.

[38] Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. *CoRR*, abs/2009.07118, 2020.

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

[40] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics, 2020.

[41] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2895–2905. Association for Computational Linguistics, 2019.

[42] Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training. *CoRR*, abs/2103.11955, 2021.

[43] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[44] Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. Entailment as few-shot learner. *CoRR*, abs/2104.14690, 2021.

[45] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[46] Wenpeng Yin, Nazneen Fatema Rajani, Dragomir R. Radev, Richard Socher, and Caiming Xiong. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8229–8239. Association for Computational Linguistics, 2020.

[47] Haiyang Yu, Ningyu Zhang, Shumin Deng, Hongbin Ye, Wei Zhang, and Huajun Chen. Bridging text and knowledge with multi-prototype embedding for few-shot relational triple extraction. In Donia Scott, Núria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6399–6410. International Committee on Computational Linguistics, 2020.

[48] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. Diverse few-shot text classification with multiple metrics. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1206–1215. Association for Computational Linguistics, 2018.

[49] Ningyu Zhang, Shumin Deng, Zhanlin Sun, Jiaoyan Chen, Wei Zhang, and Huajun Chen. Relation adversarial network for low resource knowledge graph completion. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1–12. ACM / IW3C2, 2020.

[50] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample {bert} fine-tuning. In *International Conference on Learning Representations*, 2021.

[51] Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. *CoRR*, abs/2102.09690, 2021.

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