Contrastive Demonstration Tuning for Pre-trained Language Models

Anonymous ACL submission

Abstract

Pretrained language models can be effectively stimulated by textual prompts or demonstrations, especially in low-data scenarios. Recent works have focused on automatically searching discrete or continuous prompts or optimized verbalizers, yet studies for the demonstration are still limited. Concretely, the demonstration examples are crucial for an excellent final performance of prompt-tuning. In this paper, we propose a novel pluggable, extensible, and efficient approach named contrastive demonstration tuning, which is free of demonstration sampling. Furthermore, the proposed approach can be: (i) Plugged to any previous prompt-tuning approaches; (ii) Extended to widespread classification tasks with a large number of categories. Experimental results on 16 datasets illustrate that our method integrated with previous approaches LM-BFF and P-tuning can yield better performance[1].

1 Introduction

Pre-trained language models (PLMs) have been applied to widespread natural language understanding and generation tasks, which are proven to obtain significant gains across benchmarks (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020a; Dong et al., 2019; Bao et al., 2020). One paradigm of PLMs is the pre-train—fine-tune, which has become the *de facto* standard for natural language processing (NLP), where task-specific objectives and additional parameters are leveraged in the tuning procedure. Recently, the paradigm of the adaptation of PLMs is shifting. A new fine-tuning methodology named prompt-tuning with a natural language **prompt** and a few **demonstrations** has made waves in the NLP community by proving astounding few-shot capabilities on myriad language understanding tasks. Further studies try to mitigate the labour-intensive prompt engineering with discrete prompt searching (Shin et al., 2020) or continuous prompt optimization (Liu et al., 2021d; Li and Liang, 2021; Hambardzumyan et al., 2021a; Zhong et al., 2021). However, few studies have focused on the demonstration, which is an indispensable component in prompt-oriented methodologies.

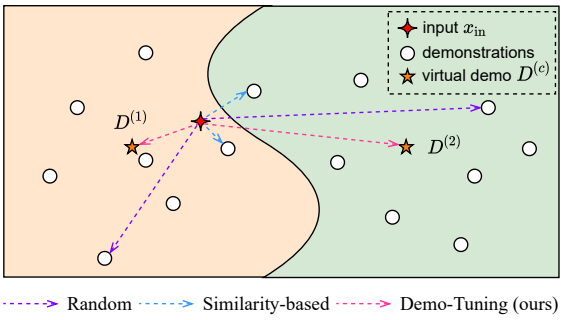

Figure 1: Comparison among current sampling strategies on demonstration-based learning. Compared to random and similarity-based sampling, demo-tuning can obtain better demonstration distributions.

In previous studies, demonstrations are sampled examples in the training set. GPT-3's naive "in-context learning" paradigm picks up to 32 randomly sampled instances as demonstrations and directly concatenates them with the input sequence (Liu et al., 2021a). Since informative demonstrations are crucial for model performance, Gao et al. (2021a) develop a refined strategy via sampling input pairs with similar examples, thereby providing the model with more discriminative comparisons. However, it is still not guaranteed to prioritize the most informative demonstrations as (1) the similarity-based sampling may obtain degraded demonstrations in different classes but have similar distances to the input; (2) the number of usable demonstrations is still bounded by the model's maximum input length. For example, as shown in Figure 1, the purple lines refer to the random sampling while the blue lines indicate similarity-based sampling. Note that similarity-based sam-

[1]Code and datasets will be released for reproducibility.

pling may obtain examples very similar to the input sequence. However, those sampled examples with different labels may tend to have a similar representation and thus confuse the discriminability of the model. Moreover, for datasets with many classes, it is still non-trivial to concatenate all sampled demonstrations. Those above-mentioned challenges hinder the applicability of demonstration in prompt-tuning.

To address those issues, in this paper, we propose contrastive **DEMO**nstration **Tuning** (Demo-tuning) for pre-trained language models. Specifically, we leverage learnable continuous embeddings (e.g., one or two learnable tokens) as virtual demonstrations to relax the maximum number of categories. We concatenate those virtual demonstrations to the input sequence; thus, our approach can be extended to a wide variety of classification tasks with many categories. To optimize those continuous embeddings, we explore a simple contrastive framework without negative pairs (Grill et al., 2020) since it is difficult to find an appropriate negative pair in semantic space for NLP. In each training batch, we randomly sample a real example and regard the virtual and real examples as positive pairs. With contrastive learning, we can obtain informative, optimized virtual demonstrations with more discriminative comparisons.

We conduct extensive experiments on 16 NLP datasets. Our contrastive demonstration tuning can yield better performance when integrated with previous prompt-based methods (e.g., LM-BFF (Gao et al., 2021a), P-tuning (Liu et al., 2021d)). Moreover, our approach can be applied to datasets with many categories and outperform baselines. Note that our approach is model-agnostic and can be plugged into lots of prompt-based methods without the effort to select suitable demonstrations. The main contributions of this study are as follows:

- We propose a pluggable, extensible, and efficient approach of contrastive demonstration tuning for pre-trained language models. To the best of our knowledge, optimizing demonstration is also a new branch of research that has not been explored in language model prompting.

- We propose virtual demonstration and leverage contrastive learning to obtain informative demonstrations and also relax the maximum number of categories in classification tasks.

- A systematic evaluation of 16 NLP datasets shows that the proposed simple-yet-effective approach contributes towards improvements across all these tasks.

2 Related Work

2.1 Prompt-tuning

With the prevalence of GPT-3 (Brown et al., 2020), prompting PLMs for few-shot learning has become a new, popular learning paradigm in natural language processing (Schick and Schütze, 2021; Tam et al., 2021; Liu et al., 2021b) and appealed to researchers. Recently, prompt-tuning has been applied to various NLP tasks, such as named entity recognition (Cui et al., 2021; Chen et al., 2021a; Zhou et al., 2021; Ma et al., 2021), entity typing (Ding et al., 2021), relation extraction (Han et al., 2021), event extraction (Hsu et al., 2021; Ye et al., 2021), and machine translation (Tan et al., 2021). Schick and Schütze (2021, 2020) propose the PET, which reformulates the NLP tasks as cloze-style questions and yields satisfactory performance. Tam et al. (2021) further propose a denser supervision object during fine-tuning to improve the PET.

Note that handcrafting a best-performing prompt is like finding a needle in a haystack, which facilitates the labor-intensive prompt engineering, Thus, recent studies (Qin and Eisner, 2021; Hambardzumyan et al., 2021b; Chen et al., 2021b) conducted in this field have been focused on automatically searching the prompts. Shin et al. (2020) propose AUTOPROMPT, which is a gradient-based method to acquire templates and label words for prompt-tuning. Wang et al. (2021) propose EFL, which reformulates the NLP task as an entailment one and turns small LMs into better few-shot learners. Additionally, Gao et al. (2020) propose LM-BFF—better few-shot fine-tuning of language models, which utilizes a generation model to obtain templates and a refined strategy for dynamically and selectively incorporating demonstrations into each context. However, it is sub-optimal for the discrete prompt searching due to the continuous nature of neural networks.

To overcome these limitations, Liu et al. (2021d,c) propose P-tuning to to automatically search prompts in the continuous space. Li and Liang (2021) propose prefix-tuning, which optimizes a sequence of continuous task-specific vectors and keeps language model parameters frozen. Lester et al. (2021a) leverage a mechanism to learn

"soft prompts" to condition frozen language models. Zhang et al. (2021) propose a differentiable prompt learning method for few-shot NLP with optimized prompt templates as well as labels. Vu et al. (2021) propose SPoT, which learns a prompt on one or more source tasks and then uses it to initialize the prompt for a target task to boost the performance across many tasks. More related works including WARP (Hambardzumyan et al., 2021a) and OPTIPROMPT (Zhong et al., 2021) also propose to leverage continuous templates, which is more effective than discrete prompt search. To conclude, most of the existing works try to obtain optimized prompts for widespread NLP tasks; however, few studies have focused on the demonstration, which is an indispensable component in prompt-oriented learning.

Our work is orthogonal to previous prompt-tuning approaches which are aimed at optimizing prompts. The major differences between virtual demonstration and continuous prompts are that: 1) they have a wholly different training strategy since continuous prompts are optimized via backpropagation with a training set while our approach utilizes contrastive learning. 2) our approach requires no external architecture (e.g., LSTM in P-tuning), thus, making it efficient and pluggable to any prompt-tuning approaches. To date, Lee et al. (2021) is the only approach that studies the demonstration and presents a simple demonstration-based learning method for named entity recognition. Apart from Lee et al. (2021), our approach focus on general NLP classification tasks. Moreover, we propose virtual demonstrations with contrastive learning strategies, which can obtain better demonstrations and also relax the maximum number of categories in datasets.

2.2 Contrastive Learning

Contrastive learning has been long considered effective in learning meaningful representations. In the early stage, Mikolov et al. (2013) propose to learn word embeddings by regarding words nearby a target word as a positive instance while others as negative. Logeswaran and Lee (2018) further generalize this approach to learn sentence representations. Recently, Kim et al. (2021) propose a contrastive learning method that makes use of a self-guidance mechanism. Yan et al. (2021) propose ConSERT, a contrastive framework for self-supervised sentence representation transfer. Giorgi et al. (2021) propose DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. Gao et al. (2021b) leverage dropout as mimimal data augmentation and propose SimCSE, a simple contrastive learning framework that greatly advances the state-of-the-art sentence embeddings.

On the other hand, contrastive learning has been also appealed to the computer vision community (Jaiswal et al., 2020; Liu et al., 2020). Chen et al. (2020) propose SimCLR: a simple framework for contrastive learning of visual representations without requiring specialized architectures or a memory bank. Chen and He (2021) observe that simple siamese networks can learn meaningful representations even using none of the negative sample pairs, large batches, and momentum encoders.

Our work is related to Grill et al. (2020), a non-contrastive self-supervised learning approach, which relies on two neural networks, referred to as online and target networks, that interact and learn from each other. However, as opposed to this approach, we utilize the encoder in the same state while Grill et al. (2020) leverage two networks in the different states. Moreover, we focus on demonstration optimization in prompt-tuning for NLP, including learning informative demonstrations and acquiring prompt temples and label tokens.

3 Preliminaries

In this work, we focus on classification tasks in the few-shot setting, including text classification and natural language understanding, where the input x_{in} is either a sentence $x_{\text{in}} = x_1$ or a pair of sentences $x_{\text{in}} = (x_1, x_2)$. Here, we let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_i^{K\times|\mathcal{Y}|}$ denote the training set of a downstream task composed of only K training examples per class, where $\mathcal{Y}$ is label space of the task. Given a pre-trained language model comprised of two stages: an encoder $f(\cdot)$ and a classifier $g(\cdot)$ [2], we encode the input x_{in} to a sequence of hidden vectors $\{\mathbf{h}_k \in \mathbb{R}^d\}$ and take the hidden vector $\mathbf{h}_{\texttt{[CLS]}} = f(x_{\text{in}})$ of `[CLS]` [3] through classifier to obtain the probability distribution $p\left(y \mid x\right) = g\left(\mathbf{h}_{\texttt{[CLS]}}\right)$ over $y \in \mathcal{Y}$.

Prompt-based Fine-tuning Prompt-based fine-tuning (Schick and Schütze, 2021; Gao et al., 2021a) is an efficient work by designing cloze-style

[2] In standard fine-tuning, the classifier is a set of randomly initialized parameters $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{Y}|\times d}$ with softmax function.

[3] For simplicity we will denote the hidden vector $\mathbf{h}_{\texttt{[CLS]}}$ of certain input x_i through encoder using $\mathbf{h}_i$.

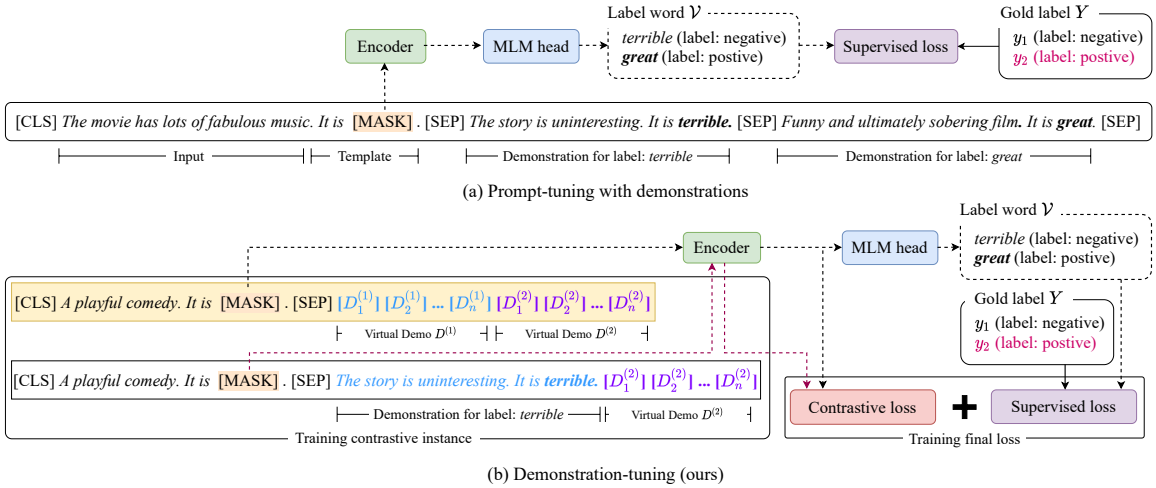

Figure 2: An illustration of (a) prompt-tuning with demonstrations, and (b) our proposed contrastive demonstration tuning (demo-tuning). Note that we regard the input with virtual demonstration and a random sampled real demonstrations as positive pairs for contrastive learning.

template $\mathcal{T}$ and verbalizer $\mathcal{M}: \mathcal{Y} \rightarrow \mathcal{V}$ mapping task labels to individual words from vocabulary $\mathcal{V}$ of pre-trained language model to fill the gap between masked LM objective of pre-trained language model and downstream fine-tuning objective.

Template In prompt-based fine-tuning paradigm, template $\mathcal{T}$ is mainly comprised of inputs x_{in} and a prompt $P = [P_i]_i^m$, where the prompt could be a series of discrete tokens (Schick and Schütze, 2021) or continual pseudo tokens (Liu et al., 2021d). For instance, in the sentiment analysis task (see Figure 2), a template with handcraft prompt may be: $\mathcal{T}(x) = \texttt{[CLS]}\, x_1, \textit{It was}\, \texttt{[MASK].[SEP]}$ where "*It was ... .*" is prompt and `[MASK]` is target which cast classification task as a language modeling task.

Verbalizer A verbalizer $\mathcal{M}$ defines a mapping of label tokens from label space of a specific task. In Figure 2a, the verbalizer maps "*negative/postive*" to "*terrible/great*". In this way, we could re-use the output weight $W_v \in \mathbb{R}^{d \times |\mathcal{V}|}$ refered *MLM head* used in pre-training and model the probability of predicting token $\mathcal{M}(y) \in \mathcal{V}$ as $p(y \mid x) = g(\mathbf{h}_{\texttt{[MASK]}})$ on hidden vector $\mathbf{h}_{\texttt{[MASK]}}$.

Demonstration Let $\mathcal{D}^c_{\text{train}}$ be the subset of all examples of class c. We sample demonstrations $d_c = (x_{\text{in}}^{(c)}, y^{(c)}) \in \mathcal{D}^c_{\text{train}}$ and convert it to $\mathcal{T}(x_{\text{in}}^{(c)}, y^{(c)})$ in which `[MASK]` is replaced by $\mathcal{M}(y^{(c)})$. We then combine the original template $\mathcal{T}$ with templates above in all classes to form $\mathcal{T}^*(x_{\text{in}})$, which will be used as a template during prompt-based tuning and inference (See Figure 2).

4 Contrastive Demonstration Tuning

In this work, we focus on how to learn a compact and differentiable **virtual demonstration** to serve as prompt augmentation instead of designing specific sampling strategies for demonstration-based learning. We propose a learning framework based on a contrastive learning approach that can be compatible with the current prompt-based learning paradigm. This section introduces the concepts of *contrastive demonstration tuning* (Demo-tuning) and provides details of this approach.

Virtual Demonstration Let $[D_i^{(c)}]_i^n$ refer to the virtual demonstration of the c^{th} class where n is a hyper-parameter to set the length of virtual demonstration, which is far less than the length of real demonstration. For instance, given a template of binary classification task (see Figure 2) as:

$$\widetilde{\mathcal{T}}(x) = \mathcal{T}(x) \oplus [D^{(1)}] \oplus [D^{(2)}] \qquad (1)$$

where $\oplus$ denotes concatenation of input sequences. $[D^{(1)}]$ and $[D^{(2)}]$ respectively denote the virtual demonstrations of two classes. Virtual demonstrations could be so flexible that can be integrated to wide variety of prompt learning approaches (Liu et al., 2021d; Lester et al., 2021b).

Next, we study how to obtain the optimal virtual demonstrations, which are initialized as a series of pseudo tokens at the start of fine-tuning. To address this challenging problem, we propose to use

contrastive learning, which aims to obtain effective representation by pulling semantically close neighbors together. Intuitively, we believe the optimal virtual demonstrations may be analogous with "prototype" (Snell et al., 2017), the representative for corresponding class, and we will discuss in §6.

Positive Instances A key element of contrastive learning is how to construct reasonable $(x_{\text{in}}, x_{\text{in}}^+)$ pairs. Here, we design a new template $\widetilde{\mathcal{T}}^+(x)$ based on template $\widetilde{\mathcal{T}}(x)$ by randomly replacing one of virtual demonstrations $[D^{(c)}]$ with real demonstration d_c as shown in the Figure 2b:

$$\widetilde{\mathcal{T}}^+(x) = \mathcal{T}(x) \oplus \mathcal{T}(x_{\text{in}}^{(1)}, y^{(1)}) \oplus [D^{(2)}] \quad (2)$$

where $[D^{(1)}]$ is replaced with a demonstration d_1 of class "*terrible*". Using this template, we could convert input x_{in} to corresponding positive example x_{in}^+, i.e., $\left(\widetilde{\mathcal{T}}(x_{\text{in}}), \widetilde{\mathcal{T}}^+(x_{\text{in}})\right)$ is a positive training instance. In this way, aligning virtual demonstration $[D^{(c)}]$ with d_c, the only difference between x_{in} and x_{in}^+, and pulling representations $(\mathbf{h}_{\text{in}}, \mathbf{h}_{\text{in}}^+)$ closer in semantic space could effectively alleviate the problem that the existing of terrible or irrelevant demonstration by previous sampling strategies.

Optimization Similar to Chen et al. (2020), we can randomly sample a minibatch of N examples from $\mathcal{D}_{\text{train}}$ to construct positive pairs $\{(x_i, x_i^+)\}_{i=1}^N$ and take a cross-entropy objective with in-batch negatives for (x_i, x_i^+):

$$\ell_i = -\log \frac{\exp(\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau)}{\sum_{j=1}^N \exp(\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau)} \quad (3)$$

where τ denotes a temperature parameter and $\text{sim}(\mathbf{h_i}, \mathbf{h_j})$ is the cosine similarity $\frac{\mathbf{h}_i^{\text{T}}\mathbf{h}_j}{\|\mathbf{h}_i\| \cdot \|\mathbf{h}_j\|}$. The negative pairs are composed of two different examples with the same demonstration in a minibatch.

In this work, we also explore a simple contrastive framework without negative pairs[4] similar to recent *non-contrastive* self-supervised learning (Grill et al., 2020). Regarding the difficulty to find a appropriate negative pair in semantic space for NLP, specially in few-shot setting, we only construct positive pairs and define the following mean squared error between $\mathbf{h}_i$ and $\mathbf{h}_i^+$ with ℓ_2-normalization,

$$\ell_i = \|\mathbf{h}_i - \mathbf{h}_i^+\|_2^2 = 2 - 2 \cdot \frac{\mathbf{h}_i^{\text{T}}\mathbf{h}_i^+}{\|\mathbf{h}_i\|_2 \cdot \|\mathbf{h}_i^+\|_2} \quad (4)$$

[4]This is the default contrastive learning method in all experiments.

where $\mathbf{h}_i$ and $\mathbf{h}_i^+$ are obtained through encoder $f(\cdot)$ in the same state different from Grill et al. (2020) which encodes x_i and x_i^+ through two networks in the different states (online network and target network).

When supervised examples $\mathcal{D}_{\text{train}}$ are available, pre-trained language model could be fine-tuned to minimize the joint objective comprised of cross-entropy and contrastive objective of Eq. (4). In this way, during inference, we can concatenate the input x_{in} with trained virtual demonstrations in template $\widetilde{\mathcal{T}}(x)$, which does not need to sample real demonstrations. Besides, we provide empirical analysis and discussion of negative sampling in §5.4.

5 Experiments

5.1 Datasets

To evaluate Demo-tuning, we conduct experiments on 6 tasks from GLUE leaderboard (Wang et al., 2019) and 10 other popular classification tasks, including natural language inference (SNLI, MNLI, QNLI, RTE), sentiment classification (SST-2, SST-5, MR, CR, MPQA), paraphrase and similarity (MRPC, QQP) and sentence classification (DBpedia, Subj, TREC, Yahoo! Answers). The detailed statistics are in Appendix A.

5.2 Settings

Evaluation During training, we follow the evaluation protocol adopted in Gao et al. (2021a) and assume a development set $\mathcal{D}_{\text{dev}}$ for model selection and hyper-parameter tuning, where the size is same with $\mathcal{D}_{\text{train}}$, i.e., $|\mathcal{D}_{\text{dev}}| = |\mathcal{D}_{\text{train}}|$. For every experiment, we measure average performance across 5 different randomly sampled $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{dev}}$ splits using a fixed set of seeds.

Hyperparameter Selection We implement our framework and reproduce P-tuning by ourselves using PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2020). The main results of LM-BFF in Table 1 are from Gao et al. (2021a). We use $\text{RoBERTa}_{\text{LARGE}}$ (Liu et al., 2019) as pretrained language model and set $K = 16$. We employ AdamW as the optimizer and set same learning rate as $1\text{e}-5$ and batch size as 8 to all tasks. For the length n of virtual demonstration per class, we select it from candidate set $\{1, 2, 3, 5\}$. Detailed template and verbalizer setting for all tasks is provided in Appendix B.

	SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)
"GPT-3" in-context learning	84.8 (1.3)	30.6 (0.9)	80.5 (1.7)	87.4 (0.8)	63.8 (2.1)	53.6 (1.0)	26.2 (2.4)
Fine-tuning	81.4 (3.8)	43.9 (2.0)	76.9 (5.9)	75.8 (3.2)	72.0 (3.8)	90.8 (1.8)	88.8 (2.1)
LM-BFF (w/ Demo)	92.6 (0.5)	**50.6 (1.4)**	86.6 (2.2)	90.2 (1.2)	**87.0 (1.1)**	92.3 (0.8)	87.5 (3.2)
P-tuning (w/ Demo)	92.7 (1.4)	47.7 (3.3)	87.5 (1.3)	90.6 (1.4)	84.3 (0.8)	91.4 (1.7)	88.1 (2.7)
Demo-tuning (LM-BFF)	**93.2 (0.4)**	50.1 (0.4)	**87.9 (0.6)**	**91.5 (0.6)**	85.9 (1.5)	**92.3 (0.6)**	90.1 (2.7)
Demo-tuning (P-tuning)	92.7 (0.6)	48.7 (2.0)	86.4 (1.1)	91.4 (0.8)	86.0 (1.6)	92.0 (0.6)	**90.7 (4.5)**
	MNLI (acc)	**MNLI-mm (acc)**	**SNLI (acc)**	**QNLI (acc)**	**RTE (acc)**	**MRPC (F1)**	**QQP (F1)**
"GPT-3" in-context learning	52.0 (0.7)	53.4 (0.6)	47.1 (0.6)	53.8 (0.4)	60.4 (1.4)	45.7 (6.0)	36.1 (5.2)
Fine-tuning	45.8 (6.4)	47.8 (6.8)	48.4 (4.8)	60.2 (6.5)	54.4 (3.9)	76.6 (2.5)	60.7 (4.3)
LM-BFF (w/ Demo)	70.7 (1.3)	72.0 (1.2)	**79.7 (1.5)**	69.2 (1.9)	68.7 (2.3)	77.8 (2.0)	69.8 (1.8)
P-tuning (w/ Demo)	71.0 (2.2)	70.8 (1.7)	78.7 (1.5)	68.2 (2.1)	**70.8 (3.0)**	75.0 (13.8)	66.6 (2.9)
Demo-tuning (LM-BFF)	71.0 (2.0)	72.8 (1.5)	78.7 (1.9)	**73.1 (1.8)**	70.0 (3.4)	**78.4 (2.3)**	**70.2 (1.7)**
Demo-tuning (P-tuning)	**71.3 (1.3)**	**73.1 (1.9)**	76.4 (1.7)	71.6 (3.0)	69.8 (4.6)	78.4 (4.4)	68.9 (2.9)

Table 1: Comparison of performance of our approach with several baselines across 14 text classification tasks in few-shot setting. We report mean (and standard deviation) results of 5 random seeds. LM-BFF (w/ Demo) and P-tuning (w/ Demo): prompt-tuning methods (LM-BFF and P-tuning) using demonstration in context with manual template used in Gao et al. (2021a). Demo-tuning (LM-BFF) and Demo-tuning (P-tuning): Our proposed approach respectively based on LM-BFF and P-tuning.

5.3 Main Results

We apply our method to two popular prompt-based tuning techniques, LM-BFF and P-tuning, and compare to a number of baselines, namely: (1) standard fine-tuning in the few-shot setting; (2) "GPT-3" in-context learning: zero-shot prediction, which concatenates prompt (e.g., randomly sampled demonstrations); (4) LM-BFF using demonstration in context with a manual template. (3) P-tuning using demonstration in context with a manual template, where we do not specifically search the optimal length of continual prompt and fixed the length m to 4 in all tasks.

In Table 1, we report the performance of the baseline approaches and our two variants. First, in-context learning could achieve comparable or even higher performance to the standard fine-tuning method and prompt-tuning methods (LM-BFF and P-tuning) using demonstration in context bring consistent improvement in a majority of tasks, which means that demonstration is worth being exploited.

Second, our approach based on two prompt-based tuning techniques could consistently outperform the vanilla methods. In detail, Demo-tuning based LM-BFF improves the average score by 0.5, compared with LM-BFF with the demonstration in an input context. More importantly, Demo-tuning is flexible and orthogonal to most fine-tuning methods. Here, for evaluating the compatibility, we combine Demo-tuning with P-tuning (Liu et al., 2021d), which could lead to a 2.3 average score improvement in total. In this work, we do not specially design template for P-tuning [5]. Although templates for P-tuning and prompt length are sub-optimal, we find that Demo-tuning with P-tuning leads to consistent gains in a majority of tasks.

	DBpedia	Yahoo!
Fine-tuning	98.2 (0.1)	66.4 (1.0)
LM-BFF	98.1 (0.2)	66.2 (1.0)
LM-BFF (w/ Demo)	-	-
P-tuning	98.2 (0.2)	67.0 (0.8)
Demo-tuning (LM-BFF)	**98.3 (0.1)**	67.9 (0.8)
Demo-tuning (P-tuning)	**98.3 (0.1)**	**68.4 (1.1)**

Table 2: Performance on multi-class sentence classification, DBpedia and Yahoo!. The size of label space $|\mathcal{Y}|$ are respectively 14 and 10. Due to sequence length limitation in pretrained language model, LM-BFF with demonstration-based learning can not be applied here.

Third, an advantage of our proposed virtual demonstration is that it could be well applied for multi-class sentence classification tasks. Table 2

[5] We simply construct template $\mathcal{T}(x)$ for P-tuning as `[CLS]` x_1 `[PROMPT][MASK][SEP]` in single-sentence tasks and `[CLS]` x_1, `[MASK]` ? x_2 `[PROMPT][SEP]` in sentence pair tasks, where `[PROMPT]` denotes continual prompt.

	SST-2	TREC	SNLI	MRPC
LM-BFF	92.7	84.8	77.2	74.5
Random	92.3	85.6	78.8	70.9
Filter-based (RoBERTa)	92.7	83.4	79.5	76.6
Filter-based (SBERT)	92.6	87.5	**79.7**	77.8
Virtual Demo (w/ Mean)	90.9	85.9	75.3	66.4
Virtual Demo (w/ CL)	**93.2**	**90.7**	78.7	**78.4**

Table 3: Impact of demonstration sampling strategies. Random: uniform sampling from each class. Filter-based: filtered sampling strategy proposed in Gao et al. (2021a) respectively based on RoBERTa and SBERT (Reimers and Gurevych, 2019). Virtual Demo (w/ mean): averaing the representations of instances with the same label as virtual demonstration.

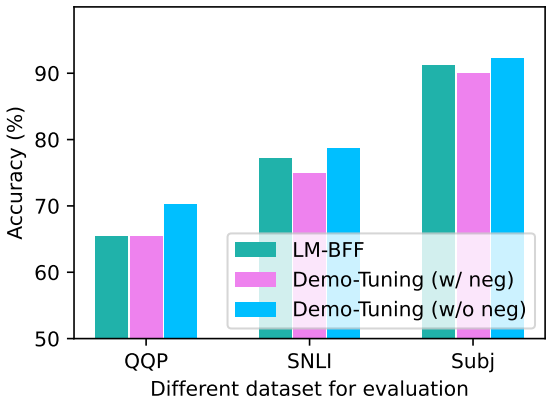

Figure 3: Ablation study on virtual demonstration optimization w/ Vs. w/o negative sampling. Demo-tuning (w/ neg): using conventional contrastive learning with negative samples to optimize virtual demonstration. Demo-tuning (w/o neg): Demo-tuning using our simplified optimization method without negative samples.

gives the results of Demo-tuning compared to standard fine-tuning and prompt-based tuning. Due to the limitation of the model's input length, in-context learning and LM-BFF with demonstration could not be applied in this scenario. We notice that while the performance of LM-BFF is worse than fine-tuning, Demo-tuning based on LM-BFF improves the score by 1.7 and achieves a better score compared to fine-tuning.

5.4 Analysis of Virtual Demonstration

The selection of demonstration is crucial for demonstration-based learning (e.g., in-context learning and LM-BFF with demonstration). Next, we compare and discuss our proposed virtual demonstration with current approaches.

Demonstration Sampling Table 3 provides the impact of demonstration sampling strategies. During inference, our proposed virtual demonstration obtained by contrastive learning during training could be as an alternative to real demonstrations, which could be viewed as an implicit sampling strategy. We compare our method with previous sampling strategies based on LM-BFF.

While the performance of uniform demonstration sampling from each class is better than the vanilla LM-BFF in TREC and SNLI, we notice that on the MRPC task, this method causes severe accuracy loss, which is up to 3.6. We think that random sampling is prone to generate irrelevant information in demonstrations. To address the above issue, Gao et al. (2021a) utilize RoBERTa or SBERT (Reimers and Gurevych, 2019) to select relevant demonstrations to examples. The filter-based sampling strategy could achieve consistent gains in the majority of tasks, which yields the highest improvement with 3.6 on the TREC task. We consider that this KNN-style method, which concatenates examples and demonstration that semantically close to example, could promote language model to decipher meaningful patterns.

Virtual demonstration, an alternative of the real demonstration during inference, i.e., avoid complex sampling step, could achieve gains in the majority of tasks. Besides our proposed method, We design a simple strategy to construct virtual demonstrations via averaging the representations of instances with the same label. We notices that constructing virtual demonstration with simple averaging of instances cause poor performance in the most tasks. However, our method with contrastive learning is more predominant than previous approaches. The only exception is SNLI, which score only is comparable with random sampling. We hypothesize that this is caused by some confusion issues, which may exist in filter-based strategy regarding semantically closeness among contrastive demonstrations.

Optimization w/ Vs. w/o Negative Samples Figure 3 gives the results of comparison between virtual demonstration optimization with negative sampling and without negative sampling. We conduct experiments with different optimization strategies on 3 tasks. We find that optimizing objective of Eq.3, i.e., conventional contrastive learning with negative samples, causes dramatically performance degradation, in which the average score is even lower than LM-BFF's. We think there are two

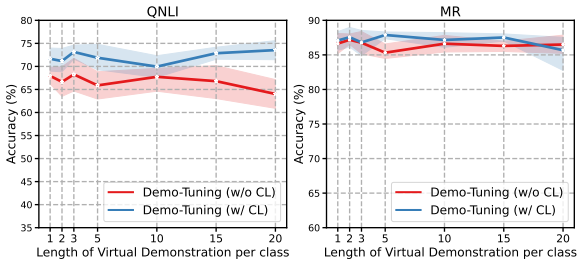

Figure 4: Ablation study on length n of virtual demonstration per class. Demo-tuning (w/o CL): Demo-tuning without contrastive learning (CL), i.e., virtual demonstration will degrade into continual prompt.

possible reasons: (1) In NLP tasks, finding a semantically reasonable negative pair is difficult, especially in the few-shot setting; (2) Negative pairs may become example-demonstrations pairs without specific limitation, which will cause a certain confusion to model. Moreover, our goal is to obtain optimal virtual demonstrations for downstream tasks. Using contrastive optimization without negative sampling may be a more suitable solution.

Demonstration Length Figure 4 shows the ablation study on length n of virtual demonstration per class. We compare Demo-tuning with its variant without contrastive learning in different settings about length n. It is noteworthy that without contrastive learning, a virtual demonstration will degrade into a continual prompt. We find that a relatively shorter length (e.g., 2 or 3) could gain stable improvement of performance in QNLI and MR. Oppositely, a larger length (e.g., 20) may decrease the performance. We consider that as the length of virtual demonstration increases, it will introduce more parameters into the model, making it challenging to learn from a small amount of annotated data. Demo-tuning could achieve consistent improvement in different lengths compared to its variant. Hence, we can conclude that **virtual demonstration optimized by simple contrastive framework plays a different role from continuous prompt**.

6 Discussion

We will discuss several favorable properties of contrastive demonstration tuning and present some open problems:

Possible Supplement for Parameter-efficient Fine-tuning. Previous studies (Liu et al., 2021d; Li and Liang, 2021) have demonstrate the effectiveness of prompt-tuning (e.g., P-tuning, Prefix-tuning) as an parameter-efficient fine-tuning methodology for huge PLMs. Our approach can serve as a supplement or parameter-efficient fine-tuning via only tuning demonstration with PLM fixed. We leave this for future works.

Relation to Prototype Learning. In §4, we have notice that the optimal virtual demonstrations may be analogous with "prototype" (Snell et al., 2017), representative for corresponding class. Our approach may have connections to prototype learning, and further empirical and theoretical analysis should be conducted.

Demonstration as External Knowledge. Recall that those concatenated demonstrations are similar to previous studies such as RAG (Lewis et al., 2020b), REALM (Guu et al., 2020) which retrieve and concatenate relevant texts as external knowledge. We think that it is also interesting to investigate novel knowledge injection approaches via demonstration.

We further discuss a few weaknesses of our method in its current form and look into some possible avenues for future work. On the one hand, our work still suffers from biased/long-tailed label distribution. Note that we obtain optimized virtual demonstration via contrastive learning; thus, those virtual demonstrations of classes with many samples may dominate the training stage.This limitation might be ameliorated with weighted sampling strategies. On the other hand, our approach cannot directly handle structure prediction tasks. Integrating demonstration with prefix-tuning-based methods may help to mitigate such limitations.

7 Conclusion and Future Work

In this work, we propose contrastive demonstration tuning, a simple model-agnostic approach for pre-trained language models, which improves state-of-the-art prompt-tuning performance without the necessity of demonstration selection.

In the future, we plan to explore the following directions: 1) studying the connection between virtual demonstration and prototypes and theoretically analyzing the optimal solution of demonstration for prompt-tuning. 2) applying our work to more NLP tasks and trying to adapt to structure prediction and natural language generation. 3) extending our work to multimodal settings and investigating demonstrations across visual and language.

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. 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.

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. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. 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 1597–1607. PMLR.

Xiang Chen, Ningyu Zhang, Lei Li, Xin Xie, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021a. Lightner: A lightweight generative framework with prompt-guided attention for low-resource ner. *arXiv preprint arXiv:2109.00720*.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021b. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. *CoRR*, abs/2104.07650.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 15750–15758. Computer Vision Foundation / IEEE.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1835–1845. Association for Computational Linguistics.

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

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *CoRR*, abs/2108.10604.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *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.

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

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

John M. Giorgi, Osvald Nitski, Bo Wang, and Gary D. Bader. 2021. Declutr: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 879–895. Association for Computational Linguistics.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap your own latent - A new approach to self-supervised learning. In *NeurIPS*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented 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 3929–3938. PMLR.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021a. WARP: word-level adversarial reprogramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4921–4933. Association for Computational Linguistics.

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

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

I Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, et al. 2021. Event extraction as natural language generation. *arXiv preprint arXiv:2108.12724*.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *CoRR*, abs/2011.00362.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2528–2540. Association for Computational Linguistics.

Dong-Ho Lee, Mahak Agarwal, Akshen Kadakia, Jay Pujara, and Xiang Ren. 2021. Good examples make A faster learner: Simple demonstration-based learning for low-resource NER. *CoRR*, abs/2110.08454.

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

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021b. The power of scale for parameter-efficient prompt tuning. In *EMNLP (1)*, pages 3045–3059. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *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.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

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

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

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

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised learning: Generative or contrastive. *CoRR*, abs/2006.08218.

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

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

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021. Template-free prompt tuning for few-shot NER. *CoRR*, abs/2109.13532.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural*

Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 3111–3119.

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. 2019. Pytorch: An imperative style, high-performance deep learning library. In *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.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

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

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *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.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *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.

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

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

Zhixing Tan, Xiangwen Zhang, Shuo Wang, and Yang Liu. 2021. MSP: multi-stage prompting for making pre-trained language models better translators. *CoRR*, abs/2110.06609.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better frozen model adaptation through soft prompt transfer. *CoRR*, abs/2110.07904.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP (Demos)*, pages 38–45. Association for Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5065–5075. Association for Computational Linguistics.

Hongbin Ye, Ningyu Zhang, Zhen Bi, Shumin Deng, Chuanqi Tan, Hui Chen, Fei Huang, and Huajun Chen. 2021. Learning to ask for data-efficient event argument extraction. *arXiv preprint arXiv:2110.00479*.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. *CoRR*, abs/2108.13161.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5017–5033. Association for Computational Linguistics.

Xin Zhou, Ruotian Ma, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021. Plug-tagger: A pluggable sequence labeling framework using language models. *CoRR*, abs/2110.07331.

A Datasets

Table 4 provides the dataset evaluated in this work.

Dataset	$\lvert\mathcal{Y}\rvert$	#Train	#Test	**Type**
SST-2	2	6,920	872	sentiment
SST-5	5	8,544	2,210	sentiment
MR	2	8,662	2,000	sentiment
CR	2	1,775	2,000	sentiment
MPQA	2	8,606	2,000	opinion polarity
Subj	2	8,000	2,000	subjectivity
TREC	6	5,452	500	question cls.
DBpedia	14	560,000	70,000	sentence cls.
Yahoo! Answers	10	1,400,000	60,000	sentence cls.
MNLI	3	392,702	9,815	NLI
SNLI	3	549,367	9,842	NLI
QNLI	2	104,743	5,463	NLI
RTE	2	2,490	277	NLI
MRPC	2	3,668	408	paraphrase
QQP	2	363,846	40,431	paraphrase

Table 4: The datasets evaluated in this work. $\lvert\mathcal{Y}\rvert$: the number of classes for classification tasks. Notes that we only sample $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{dev}}$ of $K \times \lvert\mathcal{Y}\rvert$ examples from the original training data set in our few-shot setting.

B Template settings

Table 5 and Table 6 provides manual templates and verbalizer similar with Gao et al. (2021a). We set the template of demonstration same with example.

Template	Tasks
`[CLS]` x_1 *It was* `[MASK].[SEP]`	SST-2, SST-5, MR, CR, MPQA, DBpedia, Yahoo! Answers
`[CLS]` x_1 *This is* `[MASK].[SEP]`	Subj
`[CLS] [MASK]:` x_1 `[SEP]`	TREC
`[CLS]` x_1 *?* `[MASK],` x_2 `[SEP]`	MNLI, SNLI, QNLI, RTE
`[CLS]` x_1 `[MASK],` x_2 `[SEP]`	MRPC, QQP

Table 5: Templates for all tasks evaluated in our work.

Task	Verbalizer
SST-2	incorrect/correct
SST-5	terrible/bad/okay/good/great
MR	terrible/great
CR	terrible/great
MPQA	terrible/great
Subj	subjective/objective
TREC	Description/Entity/Expression/ Human/Location/Number
DBpedia	company/institution/artist/athlete/ office/holder/transportation/building/ place/village/animal/plant/album/film/ written/work
Yahoo!	society/science/health/education/ internet/sports/business/entertainment/ family/politics

Table 6: Verbalizer for all tasks evaluated in our work.