Pre-trained Token-replaced Detection Model as Few-shot Learner

Zicheng Li **Shoushan Li*** **Guodong Zhou**
Natural Language Processing Lab, Soochow University, China
20205227019@stu.suda.edu.cn
{lishoushan, gdzhou}@suda.edu.cn

Abstract

Pre-trained masked language models have demonstrated remarkable ability as few-shot learners. In this paper, as an alternative, we propose a novel approach to few-shot learning with pre-trained token-replaced detection models like ELECTRA. In this approach, we reformulate a classification or a regression task as a token-replaced detection problem. Specifically, we first define a template and label description words for each task and put them into the input to form a natural language prompt. Then, we employ the pre-trained token-replaced detection model to predict which label description word is the most original (i.e., least replaced) among all label description words in the prompt. A systematic evaluation on 16 datasets demonstrates that our approach outperforms few-shot learners with pre-trained masked language models in both one-sentence and two-sentence learning tasks. [1]

1 Introduction

Few-shot learning aims to learn models with a few examples and the learned models generalize well from very limited examples like humans. Recently, few-shot learning has become an important and interesting research field of intelligence (Lake et al., 2015; Yogatama et al., 2019). Compared to data-rich supervised learning, few-shot learning greatly overcomes the expensive data annotation challenge in reality.

Some large pre-trained language models such as GPT-3 (Brown et al., 2020) have achieved remarkable few-shot performance by reformulating tasks as language model problems. However, its hundreds of billions of parameters deter researchers and practitioners from applying it widely. To tackle this, a new paradigm, equipping smaller masked language models (Devlin et al., 2018) with few-shot capabilities (Schick and Schütze, 2020a,b; Gao et al., 2021) has been explored, wherein downstream tasks are treated as cloze questions. Typically, as illustrated in Figure 1(b), each input sentence is appended with a prompt phrase such as "It was [MASK]" to each input sentence, allowing the model to fill in the [MASK] by reusing the masked language model head.

Instead of masked language models, another self-supervised pre-training task called token-replaced detection has been proposed by Clark et al. (2020) and it trains a model named ELECTRA to distinguish whether each token is replaced by a generated sample or not. One major advantage of token-replaced detection pre-training modeling is that it is more computationally efficient than masked language modeling. Moreover, their research demonstrates that given the same model size, pre-trained token-replaced detection models achieve substantially better performance than the pre-trained masked language model such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) in many downstream tasks.

In this paper, inspired by the above unique effectiveness of the pre-trained token-replaced detection model, we propose a new approach, pre-trained token-replaced detection models as few-shot learners, aiming to further improve the few-shot learning performances. The key idea of our approach is to reformulate downstream tasks as token-replaced detection problems. Specifically, we first define a template and label description words which will be used to convert the input sentence into a prompted text. Then, we directly insert the template and all label description words into the sentence to form a prompt that might be an ungrammatical sentence. The motivation of this operation is to make our inputs similar to those in the data for training ELECTRA, having some replaced tokens. Lastly, we use a pre-trained token-replaced detection model

*Corresponding author
[1]Our code is available at https://github.com/cjfarmer/TRD_FSL

Proceedings of the 29th International Conference on Computational Linguistics, pages 3274–3284
October 12–17, 2022.

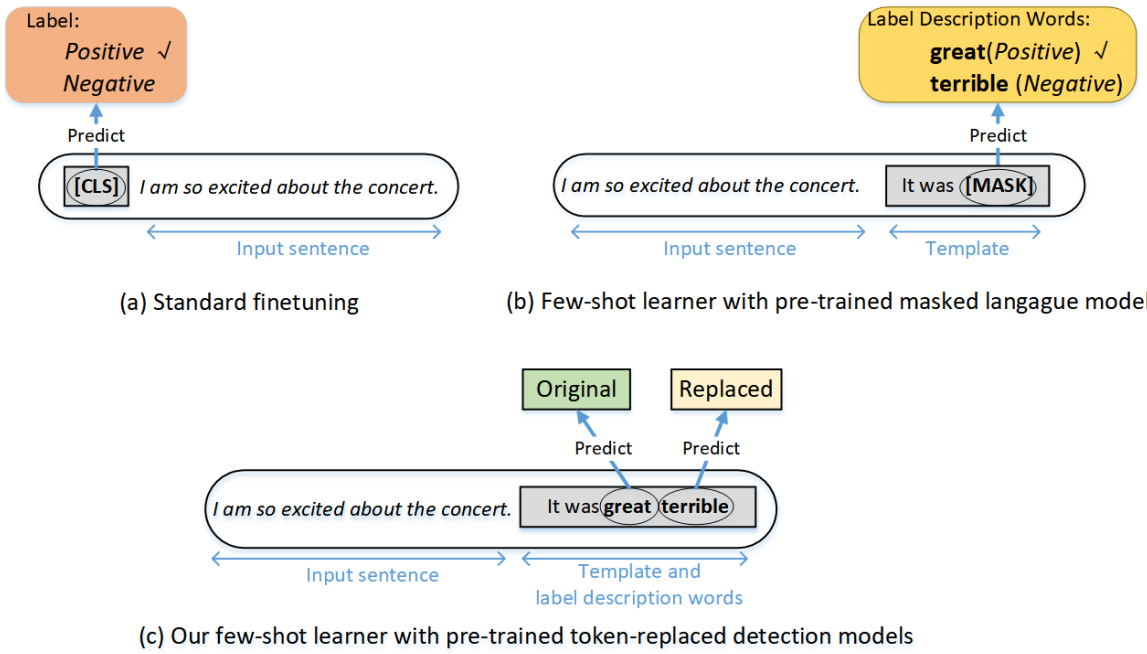

Figure 1: Different approaches of applying pre-trained models to sentiment classification.

to distinguish which label description word is the most original (i.e., least replaced) among all label description words. For instance, as illustrated in 1(c), when performing a sentiment classification task, the input sentence "*I am so excited about the concert.*" is converted into a new one "*I am so excited about the concert.* It was **great terrible**" where "**great**" and "**terrible**" are two label description words for the two sentimental categories: *positive* and *negative*. Consequently, the pre-trained token-replaced detection model may predict the label description word "**great**" is more original (i.e., less replaced) than "**terrible**", which indicates that the input belongs to a *positive* category. Compared to few-shot learners with pre-trained masked language models, in general, there are two major differences as follows. First, the designed phrases of few-shot learners with pre-trained masked don't contain any label description word. However, in our approach, we put them in prompts directly, which is easier to understand. Second, few-shot learners with pre-trained masked language models predict which label description word is the most appropriate to fill in [MASK], but our approach predicts which label description word is the most original (i.e., least replaced).

To evaluate the few-shot capacity of our approach, we use both ELECTRA-Base and ELECTRA-Large as pre-trained token-replaced detection models to perform few-shot learning in our approach and conduct experiments in a wide variety of both one-sentence and two-sentence tasks. Empirical studies demonstrate that our approach outperforms few-shot learners with pre-trained masked language models.

The contributions of this study are as follows:

- We propose a new approach for few-shot learning, which is simple and effective. To the best of our knowledge, few-shot learners with pre-trained token-replaced detection models is a novel branch of research that has not been explored in few-shot learning studies.

- A systematic evaluation of 16 popular datasets demonstrates that when given only a small number of labeled samples per class, our approach outperforms few-shot learners with pre-trained masked language models on most of these tasks.

The remainder of this paper is organized as follows. Section 2 overviews related studies about few-shot learning approaches and pre-trained token-replaced detection models. Section 3 proposes our few-shot learner with a pre-trained token-replaced detection model in detail. Section 4 presents the experimental results and analysis. Finally, Section 5 discusses the conclusions and future work.

2 Related Work

2.1 Prompt-based few-shot learning

Few-shot learning with language model prompting has arisen with the introduction of GPT-3 (Brown et al., 2020), which adds a task description (prompt) with a training example demonstration to make the language model a few-shot learner. GPT-3's naive "in-context learning" paradigms have been applied to various tasks such as text classification (Min et al., 2021; Lu et al., 2021), question answering (Liu et al., 2021a), and information extraction (Zhao et al., 2021), which shows that a large pre-trained language model can achieve remarkable performance with only a few annotated samples. However, GPT-3's dependence on gigantic pre-trained language models narrows its scope of real applications.

Instead of using a gigantic pre-trained language model, Schick and Schütze (2020a,b) reformulates a natural language processing (NLP) task as a cloze-style question with smaller masked language models (Devlin et al., 2018). Their results show that it is possible to achieve few-shot performance similar to GPT-3 with much smaller language models. Due to the instability of manually designed prompts, many subsequent studies explore automatically searching the prompts, either in a discrete space (Gao et al., 2021; Jiang et al., 2020; Haviv et al., 2021; Shin et al., 2020; Ben-David et al., 2021) or in a continuous space (Qin and Eisner, 2021; Hambardzumyan et al., 2021; Han et al., 2021; Liu et al., 2021b; Zhang et al., 2022). The discrete prompt is usually designed as natural language phrases with blank to be filled while the continuous prompt is a sequence of vectors that can be updated arbitrarily during learning. For instance, LM-BFF (Gao et al., 2021) employs pre-trained mask language models and generates discrete prompts automatically. Liu et al. (2021b) propose a prompt-based approach named P-tuning, which searches prompts in the continuous space by LSTM. Zhang et al. (2022) propose another prompt-based approach named DifferentiAble pRompT (DART), which optimizes the prompt templates and the target labels differentially.

Different from all existing above few-shot learning approaches, our approach reformulates NLP tasks as token-replaced detection problems and leverages label description words in the prompt.

2.2 Token-replaced detection

The token-replaced detection pre-training task is first introduced by Clark et al. (2020). Similar to the structure of GAN (Goodfellow et al., 2014), it pre-trains a small generator to replace some tokens in an input with their plausible alternatives and then a large discriminator to distinguish whether each word has been replaced by the generator or not. The unique effectiveness of the pre-trained token-replaced detection model intrigues many studies to apply it in many NLP tasks, such as fact verification (Naseer et al., 2021), question answering (Alrowili and Shanker, 2021; Yamada et al., 2021), grammatical error detection (Yuan et al., 2021), emotional classification (Zhang et al., 2021; Guven, 2021), and medication mention detection (Lee et al., 2020). There are also some other studies that upgrade or extend the token-replaced detection pre-training mechanism. For instance, Meng et al. (2021) jointly train multiple generators of different sizes to provide training signals at various levels of difficulty. Futami et al. (2021) transfer the mechanism to visual pre-training and Fang et al. (2022) propose an extended version of ELECTRA for speech recognition.

Different from all the above studies, to the best of our knowledge, this paper is the first study to apply pre-trained token-replaced detection models to few-shot learning.

3 Our approach

A pre-trained token-replaced detection model like ELECTRA (Clark et al., 2020) trains a discriminator $\mathcal{D}$ that detects whether a token x_t in an input token sequence $x = [x_1, \ldots, x_t, \ldots, x_n]$ is an original or replaced one. Suppose that the output from the discriminator is $y = [y_1, \ldots, y_t, \ldots, y_n]$. Then, $y_t = 0$ (or 1) indicates that x_t (at the position t) is an original (or a replaced) token. Specifically, in ELECTRA (Clark et al., 2020), the discriminator is trained together with a masked language modeling generator which is used to generate replaced tokens in a token sequence. Finally, the discriminator performs the prediction with a sigmoid output layer, i.e.,

$$P\left(y_t \mid x_t\right) = \text{sigmoid}\left(w^T h_D\left(x_t\right)\right) \quad (1)$$

where $h_D\left(x_t\right)$ is the encoder function in the discriminator $\mathcal{D}$.

3.1 Few-shot Classification

3.1.1 Few-shot Fine-tuning Phase

Suppose that the downstream classification task is a one-sentence classification problem and it has k labels with label space Y where $|Y| = k$. For the i-th category, we hand-craft a label description word $\mathbf{LABEL(i)}$. Then an input x is rewritten as a prompt as follows:

$$x_{\text{prompt}} = x \text{ It was } \mathbf{LABEL(1)} \ldots \mathbf{LABEL(i)} \ldots \mathbf{LABEL(k)} \quad (2)$$

When the downstream classification task is a two-sentence classification problem, the input (x_1, x_2) is rewritten as a prompt as follows:

$$x_{\text{prompt}} = < x_1 > ? \mathbf{LABEL(1)} \ldots \mathbf{LABEL(i)} \ldots \mathbf{LABEL(k)}, < x_2 > \quad (3)$$

Suppose that this sample belongs to the i-th label and the positions of the label description words are $[t_1, \ldots, t_i, \ldots, t_k]$. Thus, the output of the i-th description word y_{t_i} is set to be 0 and this label description word is considered as original. In contrast, the outputs of all the other label description words are set to be 1 and these description words are considered as replaced. Note that the outputs of all tokens beyond label description words are set to be 0. Formally, the output of the whole prompt is obtained as follows:

$$y_{\text{prompt}} = [\ldots, y_{t_1-1} = 0, y_{t_1} = 1, \ldots, y_{t_i} = 0, \ldots, y_{t_k} = 1, y_{t_k+1} = 0, \ldots] \quad (4)$$

For instance, in a 5-category sentiment classification task, an input x ="*This is one of his best films.*" could be rewritten as x_{prompt} = "*This is one of his best films.* It was **great good okay bad terrible**" where "**great**", "**good**", "**okay**", "**bad**", and "**terrible**" are used as the label description words for the *very positive*, *positive*, *neutral*, *negative*, and *very negative* category. The output of x_{prompt} becomes $y_{\text{prompt}} = [\ldots, 0, 0, 1, 1, 1, 1]$.

All prompt samples, together with their labels are used to update the parameters in the discriminator $\mathcal{D}$ of the pre-trained token-replaced detection model. Specifically, following the original progress of pre-training a token replaced model, we train the discriminator $\mathcal{D}$ by minimizing the binary cross entropy loss. It is important to note that our approach reuses the pre-trained weights w^T in the formula (1) and does not use any other new parameters.

3.1.2 Testing Phase

In the testing phase, a testing sample is rewritten as a prompt according to formula (2) or (3) and the labels of all label description words in this prompt is predicted with the following formula, i.e.,

$$P(y \mid \mathbf{LABEL(i)}) = \text{sigmoid}\left(w^T h_D(\mathbf{LABEL(i)})\right) \quad (5)$$

Then, the real label of the sample, i.e., l_{test}, is determined by the following formula, i.e.,

$$l_{\text{test}} = \underset{i}{\text{argmax}}(P(y = 0 \mid \mathbf{LABEL(i)})) \quad (6)$$

3.2 Few-shot Regression

3.2.1 Few-shot Fine-tuning Phase

Suppose that the downstream task is a regression problem and it has label space Y where Y is a bounded interval $[v_l, v_u]$. Following Gao et al. (2021), we reformulate the problem as a "binary classification"—predicting the probabilities of belonging to two opposing poles, $\{c_l, c_u\}$ with values v_l and v_u respectively.

Then, a few-shot regression problem can be handled as a few-shot classification problem that has two labels with label space $\{c_l, c_u\}$. Same as classification tasks above, we rewrite an input x as a prompt for a one-sentence regression task as follows:

$$x_{\text{prompt}} = x \text{ It was } \mathbf{LABEL}(l)\mathbf{LABEL}(u) \quad (7)$$

When the downstream task is a two-sentence regression task, we rewrite an input x as a prompt as follows:

$$x_{\text{prompt}} = < x_1 > \mathbf{LABEL}(l)\mathbf{LABEL}(u), < x_2 > \quad (8)$$

where $\mathbf{LABEL}(l)$ and $\mathbf{LABEL}(u)$ denote the label description words for the low and upper bound categories.

Suppose that the positions of the two label description words are $[t_l, t_u]$. Then, the output of the whole prompt is obtained as follows:

$$y_{\text{prompt}} = [\ldots, y_{t_l-1} = 0, y_{t_l} = (1 - P(c_l \mid x)), y_{t_u} = (1 - P(c_u \mid x)), y_{t_u+1} = 0, \ldots] \quad (9)$$

where $P(c_l \mid x)$ and $P(c_u \mid x)$ are the posterior probabilities of x belonging to c_l and c_u and satisfy

Category	Dataset	$\|Y\|$	#Train	#Test	Type
One-sentence	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 classification
	CoLA	2	8,551	1,042	acceptability
Two-sentence	MNLI	3	392,702	9,815	natural language inference
	MNLI-MM	3	392,702	9,832	natural language inference
	SNLI	3	549,367	9,842	natural language inference
	QNLI	2	104,743	5,463	natural language inference
	RTE	2	2,490	277	natural language inference
	MRPC	2	3,668	408	paraphrase
	QQP	2	363,846	40,431	paraphrase
	STS-B	R	5,749	1,500	sentence similarity

Table 1: The details of 16 datasets: $|Y|$: # of classes for classification tasks (Note that STS-B is a regression task over a bounded interval [0, 5]). In our few-shot experiments, we train and develop on limited examples sampled from the original training set and evaluate on the complete test set.

the equation, i.e., $P(c_l \mid x) + P(c_u \mid x) = 1$. Following Gao et al. (2021), these two probabilities could be estimated as follows:

$$P(c_l \mid x) = \frac{v_u - y}{v_u - v_l} \tag{10}$$

$$P(c_u \mid x) = \frac{y - v_l}{v_u - v_l} \tag{11}$$

For instance, in a two-sentence similarity regression task over the interval [0, 5], the label value of two sentences "*Kittens are eating food.*" and "*Kittens are eating from dishes.*" is 4.0. We use **No** and **Yes** as the label description words for the low and upper bound categories. According to formula (8-11), we construct its x_{prompt} as "*Kittens are eating food.* **No Yes**, *Kittens are eating from dishes.*" and obtain its output $y_{\text{prompt}} = [\ldots, 0, \mathbf{0.8}, \mathbf{0.2}, 0, \ldots]$.

Same as classification tasks, we also adopt binary cross entropy loss and utilize all prompt samples together with their labels to fine-tune the discriminator $\mathcal{D}$. It is important to note that our approach reuses the pre-trained weights w^T in the formula (1) and does not use any other new parameters.

3.2.2 Testing Phase

In the testing phase, a testing sample x_{test} is rewritten as a prompt according to formula (7) or (8) and the outputs of the few-shot learner is obtained with the following formula, i.e.,

$$y_l = \text{sigmoid}\left(w^T h_D(\mathbf{LABEL}(l))\right) \tag{12}$$

$$y_u = \text{sigmoid}\left(w^T h_D(\mathbf{LABEL}(u))\right) \tag{13}$$

From formula (9), we can get

$$P(c_l \mid x_{\text{test}}) = 1 - y_l \tag{14}$$

$$P(c_u \mid x_{\text{test}}) = 1 - y_u \tag{15}$$

Note that these two posterior probabilities might not satisfy the equation, i.e., $P(c_l \mid x) + P(c_u \mid x) = 1$. Therefore, we use a normalization method to update the two probabilities, i.e.,

$$P'(c_l \mid x_{\text{test}}) = \frac{P(c_l \mid x_{\text{test}})}{((P(c_l \mid x_{\text{test}}) + P(c_u \mid x_{\text{test}}))} \tag{16}$$

$$P'(c_u \mid x_{\text{test}}) = \frac{P(c_u \mid x_{\text{test}})}{((P(c_l \mid x_{\text{test}}) + P(c_u \mid x_{\text{test}}))} \tag{17}$$

Then, the regression value of the test sample, i.e., v_{test} , is obtained by using the following formula (Gao et al., 2021):

$$v_{\text{test}} = v_l \cdot P'(c_l \mid x_{\text{test}}) + v_u \cdot P'(c_u \mid x_{\text{test}}) \tag{18}$$

Task	Template	Label Space	**Label(1) . . . Label(k)**
One-sentence			
SST-2	<*S1*> It was **Label(1) . . . Label(k)**	*positive, negative*	**great, terrible**
SST-5	<*S1*> It was **Label(1) . . . Label(k)**	*very positive, positive, neutral, negative, very negative*	**great, good, okay, bad, terrible**
MR	<*S1*> It was **Label(1) . . . Label(k)**	*positive, negative*	**great, terrible**
CR	<*S1*> It was **Label(1) . . . Label(k)**	*positive, negative*	**great, terrible**
MPQA	<*S1*> It was **Label(1) . . . Label(k)**	*positive, negative*	**great, terrible**
Subj	<*S1*> This is **Label(1) . . . Label(k)**	*subjective, objective*	**subjective, objective**
TREC	**Label(1) . . . Label(k)**: <*S1*>	*abbreviation, entity, description, human, location, numeric*	**Expression, Entity, Description, Human, Location, Number**
COLA	<*S1*> This is **Label(1) . . . Label(k)**	*grammatical, not_grammatical*	**correct, incorrect**
Two-sentence			
MNLI	<*S1*> ? **Label(1) . . . Label(k)**, <S2>	*entailment, neutral, contradiction*	**Yes, Maybe, No**
MNLI-MM	<*S1*> ? **Label(1) . . . Label(k)**, <S2>	*entailment, neutral, contradiction*	**Yes, Maybe, No**
SNLI	<*S1*> ? **Label(1) . . . Label(k)**, <S2>	*entailment, neutral, contradiction*	**Yes, Maybe, No**
QNLI	<*S1*> ? **Label(1) . . . Label(k)**, <S2>	*entailment, not_entailment*	**Yes, No**
RTE	<*S1*> ? **Label(1) . . . Label(k)**, <S2>	*entailment, not_entailment*	**Yes, No**
MRPC	<*S1*> **Label(1) . . . Label(k)**, <S2>	*equivalent, not_ equivalent*	**Yes, No**
QQP	<*S1*> **Label(1) . . . Label(k)**, <S2>	*equivalent, not_ equivalent*	**Yes, No**
STS-B	<*S1*> **Label(1) . . . Label(k)**, <S2>	*[0,5]*	**Yes, No**

Table 2: Manual templates and label description words in our experiments.

4 Experiments

In this section, we compare our approach with a few-shot learning approach based on pre-trained masked language models. Furthermore, we evaluate the impact of different templates, label description words and training data scales.

4.1 Evaluation Setting

We conduct a systematic empirical study based on the datasets used in Gao et al. (2021). The experimental data contains 16 datasets from many kinds of NLP tasks such as sentiment analysis, question classification, opinion polarity, subjectivity, acceptability, natural language inference, paraphrase, and sentence similarity (Wang et al., 2018; Bowman et al., 2015). Following Gao et al. (2021), we divide these tasks into two categories, i.e., one-sentence (single sentence) input and two-sentence (sentence pair) input tasks. In addition, these tasks not only contain binary or multi-class classification but also contain regression. See the statistics of datasets in Table 1.

4.2 Evaluation protocol

Note that the results of few-shot learning experiments are very sensitive and unstable to the different splits of data and hyper-parameter setups (Dodge et al., 2020; Zhang et al., 2020), because the size of the training examples is so small. Thus, we follow the evaluation protocol of (Gao et al., 2021) by running 5 experiments with 5 different training and development splits, randomly sampled from the original training set using a fixed set of seeds, and then measuring the average results and standard deviations. Note that, following (Gao et al., 2021), we sample the same size of development set as the training set. For the hyper-parameters, we also utilize grid search to get the best hyper-parameter setup. We set the weight_decay to be 2e-3, max_length to be 256 and use AdamW optimizer with epsilon 1e-8. We change the learning rate in the set of {1e-5,2e-5,3e-5,4e-5,5e-5} and the batch size between 4 or 8. Besides, we use manual templates and label description words for each task and the details are shown in Table 2.

4.3 Main results

We use 16 samples per class for few-shot learning experiments and conduct our experiments on both base-level and large-level pre-trained model scenarios. We compare our approach with several baselines including 1) Fine-tuning: standard fine-tuning of pre-trained models; 2) P-tuning (Liu et al., 2021b): few-shot learner that searches prompts in a continuous space by LSTM; 3) LM-BFF (Gao et al., 2021): few-shot learner that employs pre-trained mask language models and discrete prompts. For a fair comparison, we use the same templates and label description words as our approach and do not use any demonstrations; 4) DART (Zhang et al., 2022): few-shot learner that optimizes the prompt templates and the target labels differentially.

One-sentence	SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)	CoLA (matt)	*AVG*
Fine-tuning(RoBERTa)	77.8 (2.8)	38.5 (1.6)	70.1 (4.9)	76.7 (2.8)	70.1 (8.0)	89.7 (0.8)	81.5 (4.3)	18.9 (11.7)	*65.4*
Fine-tuning(ELECTRA)	82.8 (3.5)	41.6 (3.6)	73.9 (3.5)	82.9 (4.1)	70.7 (5.1)	**92.0 (0.5)**	78.5 (5.8)	**39.3 (3.4)**	*70.2*
P-tuning(RoBERTa)	83.3 (5.3)	43.1 (2.1)	81.7 (1.2)	86.0 (3.6)	74.0 (5.2)	89.0 (1.1)	76.9 (8.3)	-0.8 (2.5)	*66.7*
LM-BFF(RoBERTa)	87.2 (1.3)	44.5 (0.8)	83.4 (1.4)	89.1 (1.5)	81.3 (3.9)	89.3 (1.8)	77.5 (6.1)	5.3 (5.3)	*69.7*
DART(RoBERTa)	88.9 (0.5)	45.3 (1.5)	83.7 (1.0)	89.2 (1.4)	76.6 (6.3)	88.9 (2.2)	77.3 (7.2)	4.2 (5.0)	*69.3*
Ours(ELECTRA)	**91.7 (0.8)**	**49.7 (1.0)**	**86.8 (2.8)**	**90.8 (1.0)**	**84.5 (1.5)**	87.5 (1.2)	**82.2 (3.3)**	24.7 (11.8)	***74.7***
Two-sentence	MNLI (acc)	MNLI-MM (acc)	SNLI (acc)	QNLI (acc)	RTE (acc)	MRPC (f1)	QQP (f1)	*STS-B (pear)*	*AVG*
Fine-tuning(RoBERTa)	38.6 (2.5)	39.5 (2.7)	48.0 (4.7)	**63.2 (6.7)**	51.9 (1.6)	74.5 (4.4)	58.6 (6.0)	*65.2 (8.7)*	*54.9*
Fine-tuning(ELECTRA)	46.9 (3.6)	48.9 (3.8)	50.6 (2.1)	59.9 (2.3)	52.6 (2.5)	**76.9 (2.9)**	**64.1 (2.8)**	***72.4 (2.0)***	*59.0*
P-tuning(RoBERTa)	50.6 (1.1)	50.6 (1.1)	55.0 (4.3)	58.1 (3.1)	56.0 (4.2)	70.2 (2.3)	58.7 (2.8)	-	*57.0*
LM-BFF(RoBERTa)	59.1 (2.4)	60.9 (2.4)	64.3 (2.9)	61.8 (4.8)	57.9 (6.7)	72.3 (6.6)	62.7 (2.1)	*68.6 (5.7)*	*63.5*
DART(RoBERTa)	55.3 (2.4)	55.3 (2.4)	62.6 (2.6)	58.4 (4.3)	58.2 (6.0)	72.4 (2.5)	60.4 (1.7)	-	*60.4*
Ours(ELECTRA)	**59.7 (2.4)**	**61.8 (2.0)**	**68.9 (3.2)**	61.9 (2.4)	**61.5 (2.9)**	73.9 (3.9)	58.0 (3.8)	*66.6 (2.9)*	***64.0***

Table 3: Experimental results of different approaches when base pre-trained models are used.

4.3.1 RoBERTa-Base and ELECTRA-Base Results

Table 3 gives the experimental results of different prompt-based approaches to few-shot learning with a base pre-trained model, i.e., RoBERTa-Base or ELECTRA-Base. The best performance in each task is bold in the table. Note that since there is no implementation for regressions tasks in the two baseline approaches, i.e., P-tuning and DART and thus we do not reproduce their approach on STS-B which is a regression task. From this table, we discuss the results in two scenarios, i.e., one-sentence and two-sentence tasks.

In one-sentence tasks, first, fine-tuning with RoBERTa-Base performs worse than fine-tuning with ELECTRA-Base on average (65.4% vs. 70.2%), which indicates that ELECTRA-Base is a better fine-tuner even when only a few training samples are available. This result is consistent with the conclusion reported in Clark et al. (2020) when many training samples are available. Second, all prompt-based approaches greatly outperform standard fine-tuning on most tasks, which indicates that few-shot learners with either base masked language model or base token-replaced detection model are powerful in few-shot learning. One big exception is CoLA (Warstadt et al., 2019) where few-shot learning approaches perform much worse than fine-tuning approaches. This might be because the task aims to detect whether a sentence is grammatical or non-grammatical which is difficult to find suitable label description words. However, interestingly, we find that ELECTRA-Base performs much better than RoBERTa-Base in this task. Third, our approach yields excellent results and performs much better than P-tuning, LM-BFF and DART on average (74.7% vs. 66.7%, 69.7% and 69.3%), which encourages using a pre-trained token-replaced detection model for few-shot learning in one-sentence tasks.

In two-sentence tasks, first, standard fine-tuning with RoBERTa-Base still performs worse than fine-tuning with ELECTRA-Base. Second, all prompt-based approaches greatly outperform standard fine-tuning on most tasks, which once again indicates that few-shot learners with either mask language model or token-replaced detection model are powerful in few-shot learning. Third, our approach performs better than P-tuning, LM-BFF and DART, although the average improvements are quite limited (64.0% vs. 57%, 63.5% and 60.4%).

4.3.2 RoBERTa-Large and ELECTRA-Large Results

Table 4 gives the experimental results of different prompt-based approaches to few-shot learning with a large pre-trained model, i.e., RoBERTa-Large or ELECTRA-Large. The best performance in each task is bold in the table. From this table, we discuss the results in two scenarios, i.e., one-sentence and two-sentence tasks.

In one-sentence tasks, first, fine-tuning with RoBERTa-Large performs a bit better than fine-tuning with ELECTRA-Large on average (68.1% vs. 70.4%), which indicates that the choice of ELECTRA and RoBERTa might depend on the tasks when large models are used. Second, all prompt-based approaches greatly outperform standard fine-tuning on most tasks, which indicates that few-shot learners with either large masked language models or large token-replaced detection models are powerful in few-shot learning. However, CoLA is still the exception and even

One-sentence	SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)	CoLA (matt)	*AVG*
Fine-tuning(RoBERTa)	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)**	**33.9 (14.3)**	*70.4*
Fine-tuning(ELECTRA)	79.9 (7.9)	41.2 (1.9)	73.0 (5.4)	75.0 (6.4)	65.3 (6.9)	**94.0 (1.0)**	82.8 (8.0)	33.4 (10.4)	*68.1*
P-tuning(RoBERTa)	89.6 (2.6)	48.0 (1.3)	85.4 (1.9)	88.7 (2.6)	76.3 (3.3)	90.9 (1.5)	86.2 (3.4)	4.0 (5.3)	*71.1*
LM-BFF(RoBERTa)	92.7 (0.9)	47.4 (2.5)	87.0 (1.2)	90.3 (1.0)	**84.7 (2.2)**	91.2 (1.1)	84.8 (5.1)	9.3 (7.3)	*73.4*
DART(RoBERTa)	91.6 (1.0)	47.4 (3.3)	85.7 (3.0)	90.3 (0.8)	66.6 (6.4)	89.9 (1.7)	84.8 (4.6)	10.0 (8.4)	*70.8*
Ours(ELECTRA)	**92.8 (0.6)**	**50.7 (2.9)**	**89.4 (0.8)**	**90.5 (2.2)**	83.2 (1.4)	92.1 (0.7)	87.2 (3.8)	16.3 (15.1)	***75.3***
Two-sentence	MNLI (acc)	MNLI-MM (acc)	SNLI (acc)	QNLI (acc)	RTE (acc)	MRPC (f1)	QQP (f1)	*STS-B (pear)*	*AVG*
Fine-tuning(RoBERTa)	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)	*53.5 (8.5)*	*55.9*
Fine-tuning(ELECTRA)	54.4 (2.4)	56.7 (1.7)	58.8 (4.8)	62.9 (4.1)	53.8 (3.7)	**78.7 (3.1)**	67.2 (3.4)	***78.5 (0.5)***	*63.9*
P-tuning(RoBERTa)	59.7 (3.0)	59.7 (3.0)	71.8 (3.5)	62.5 (6.5)	61.8 (2.6)	72.7 (7.4)	64.2 (1.5)	-	*64.6*
LM-BFF(RoBERTa)	68.3 (2.3)	70.5 (1.9)	77.2 (3.7)	64.5 (4.2)	69.1 (3.6)	74.5 (5.3)	65.5 (5.3)	*71.0(7.0)*	*70.1*
DART(RoBERTa)	67.1 (2.6)	67.0 (2.5)	74.0 (4.0)	63.1 (3.0)	64.5 (5.2)	75.9 (4.7)	63.4 (4.4)	-	*67.9*
Ours(ELECTRA)	**69.2 (4.0)**	**71.0 (3.5)**	**79.3 (3.2)**	**69.0 (4.5)**	**74.2 (3.1)**	73.2 (7.5)	**68.2 (3.4)**	*74.7 (2.9)*	***72.4***

Table 4: Experimental results of different approaches when large pre-trained models are used.

worse, the performance of few-shot learning with ELECTRA-Large performs worse than ELECTRA-Base, (16.3% vs. 24.7%). This result shows that the prompting style in our few-shot learning approach seems not suitable for the task of grammatical or non-grammatical detection. Third, our approach yields performances better than P-Tuning, LM-BFF and DART, achieving 4.2%, 1.9% and 4.5% average improvements respectively.

In two-sentence tasks, first, fine-tuning with RoBERTa-Large performs much worse than fine-tuning with ELECTRA-Large (55.9% vs. 63.9%). Second, all prompt-based approaches greatly outperform standard fine-tuning on many tasks, which once again indicates that few-shot learners with either mask language model or pre-trained token-replaced detection model are powerful in few-shot learning. Third, our approach performs better than P-Tuning, LM-BFF and DART on average (72.4% vs. 64.6%, 70.1% and 67.9%).

4.4 Impact of templates and label description words

We further conduct experiments on the one-sentence task SST-2 and the two-sentence task MNLI to study the impact of different templates and label description words in our approach. Due to a large number of trials in the grid search, we use a fixed batch size 4 and learning rate 2e-5 in this part. Table 5 shows the results of the LM-BFF approach with RoBERTa-Base, the best-performed approach in all prompt-based baselines, and our approach with ELECTRA-Base in the tasks of SST-2 and MNLI. From this table, we can see that the impact of different templates and label description words for our method is similar to LM-BFF. In terms of label description words, the more semantic-related the designed label words are to the categories, the more likely to achieve stable and excellent results. For instance, in SST-2, regardless of LM-BFF or our approach, the semantic-related label description words **great/terrible** and **good/bad** always outperform the words **dog/cat** and **terrible/great** which are semantically irrelevant or even opposite with the categories *positive* and *negative*. In terms of templates, the performance is a bit sensitive to the templates, even a punctuation mark. Besides, there seems to be no general principle to design templates to optimally adapt to our approach and LM-BFF. For instance, In MNLI, LM-BFF obtains the best performance with the template *"<S1>.* **Label(1) ... Label(k),** *<S2>"*, while our approach obtains the best performance with the template *"<S1>?* **Label(1) ... Label(k),** *<S2>"*.

4.5 Impact of training data scales

We further conduct experiments on the one-sentence task SST-2 and the two-sentence task MNLI to study the impact of the numbers of labeled instances in our approach. In this part, we also use a fixed batch size 4 and learning rate 2e-5. Figure 2 shows the trends of the LM-BFF approach, the best-performed approach in all prompt-based baselines, and our approach when using different numbers of labeled instances. From this figure, we can see that our approach outperforms LM-BFF in different numbers of labeled instances in the one-sentence task SST-2. In the two-sentence task MNLI, our approach performs similarly to LM-BFF when the numbers of labeled instances are less than 64. But our approach outperforms LM-BFF when the numbers of labeled instances are

Task	Template	**Label(1) . . . Label(k)**	LM-BFF (acc)	Our approach (acc)
SST-2 (*positive/negative*)	<*S1*> It was **Label(1) . . . Label(k)**	**great, terrible**	88.6 (1.3)	91.4 (1.6)
		good, bad	88.9 (0.6)	91.0 (2.0)
		dog, cat	85.2 (2.0)	79.6 (7.3)
		terrible, great	82.4 (3.3)	89.2 (1.9)
	Label(1) . . . Label(k) : <*S1*>	**great, terrible**	85.6 (3.0)	91.1 (1.2)
		good, bad	87.5 (0.4)	90.8 (0.7)
		dog, cat	80.1 (3.5)	69.7 (8.2)
		terrible, great	67.4 (3.5)	76.4 (9.6)
MNLI (*entailment/neutral /contradiction*)	<*S1*>? **Label(1) . . . Label(k)**, <*S2*>	**Yes, Maybe, No**	58.3 (2.4)	58.8 (2.5)
	<*S2*>. **Label(1) . . . Label(k)**, <*S1*>		58.7 (1.3)	57.6 (2.5)
	<*S1*> **Label(1) . . . Label(k)** <*S2*>		56.4 (1.8)	53.6 (2.2)
	<*S1*>. **Label(1) . . . Label(k)**, this is good, <*S2*>		54.0 (2.4)	55.8 (3.5)

Table 5: The impact of different templates and label description words.

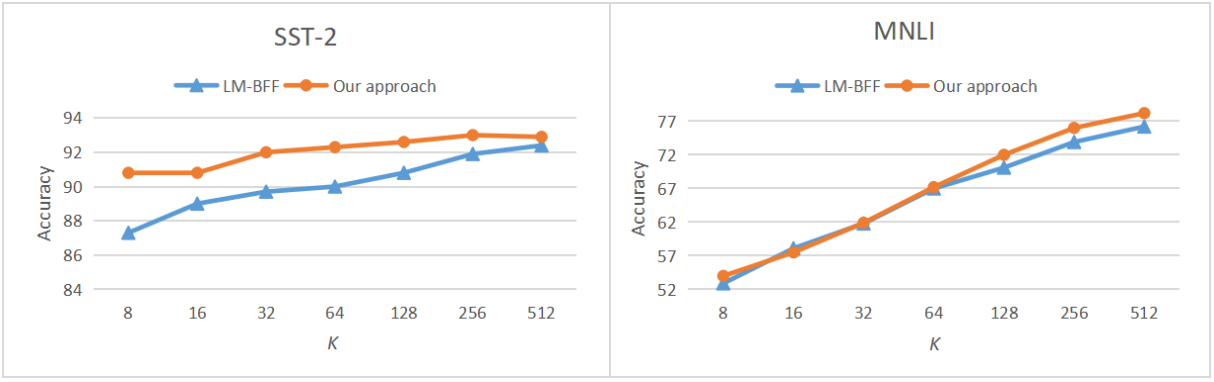

Figure 2: LM-BFF vs. our approach when using different numbers of labeled instances (*K*: # of labeled instances per class).

among [128, 512].

5 Conclusion and Future Work

In this paper, we propose a novel few-shot learning approach with pre-trained token-replaced detection models, which transforms traditional classification and regression tasks into token-replaced detection problems. Empirical studies on 16 NLP datasets demonstrate that, in both one-sentence and two-sentence learning tasks, our approach generally achieves better performances in the few-shot scenario when compared to the masked language model-based few-shot learner. These results highlight that our approach is a comprehensive alternative for few-shot learning.

In the future, we would like to explore the following directions. First, we notice that in some tasks like CoLA, standard fine-tuning is also a strong baseline and even performs much better than few-shot learners based on either a masked language model or a token-replaced detection model. Thus, it is interesting to combine [CLS] output vector, i.e., the standard fine-tuning style, with the prompting style, to further improve the few-shot learning performance. Second, we would like to apply our approach to some other NLP tasks, such as multi-label text classification and sequence labeling tasks like named entity recognition.

Acknowledgement

This work was supported by a NSFC grant (No.62076176). We also acknowledge reviewers for their valuable suggestions.

References

Sultan Alrowili and K Shanker. 2021. Large biomedical question answering models with albert and electra. *CLEF (Working Notes)*.

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. Pada: A prompt-based autoregressive approach for adaptation to unseen domains. *arXiv preprint arXiv:2102.12206*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

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

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. 2022. Corrupted image modeling for self-supervised visual pre-training. *arXiv preprint arXiv:2202.03382*.

Hayato Futami, Hirofumi Inaguma, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. 2021. Asr rescoring and confidence estimation with electra. *arXiv preprint arXiv:2110.01857*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. 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 (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Zekeriya Anil Guven. 2021. The effect of bert, electra and albert language models on sentiment analysis for turkish product reviews. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pages 629–632. IEEE.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. Warp: Word-level adversarial reprogramming. *arXiv preprint arXiv:2101.00121*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. Bertese: Learning to speak to bert. *arXiv preprint arXiv:2103.05327*.

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

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Lung-Hao Lee, Po-Han Chen, Hao-Chuan Kao, Ting-Chun Hung, Po-Lei Lee, and Kuo-Kai Shyu. 2020. Medication mention detection in tweets using electra transformers and decision trees. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 131–133.

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

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv: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. *arXiv preprint arXiv:1907.11692*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Yu Meng, Chenyan Xiong, Payal Bajaj, Paul N Bennett, Jiawei Han, Xia Song, et al. 2021. Pretraining text encoders with adversarial mixture of training signal generators. In *International Conference on Learning Representations*.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Noisy channel language model prompting for few-shot text classification. *arXiv preprint arXiv:2108.04106*.

Muchammad Naseer, Muhamad Asvial, and Riri Fitri Sari. 2021. An empirical comparison of bert, roberta, and electra for fact verification. In *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 241–246. IEEE.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*.

Timo Schick and Hinrich Schütze. 2020a. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Timo Schick and Hinrich Schütze. 2020b. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. *arXiv preprint arXiv:2106.00882*.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. Multi-class grammatical error detection for correction: A tale of two systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Differentiable prompt makes pre-trained language models better few-shot learners. In *International Conference on Learning Representations*.

Shunxiang Zhang, Hongbin Yu, and Guangli Zhu. 2021. An emotional classification method of chinese short comment text based on electra. *Connection Science*, pages 1–20.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*.

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