Discriminative Language Model as Semantic Consistency Scorer for Prompt-based Few-Shot Text Classification

Zhipeng Xie, Yahe Li
School of Computer Science, Fudan University, Shanghai, China
xiezp@fudan.edu.cn, yaheli21@m.fudan.edu.cn

Abstract

A successful prompt-based finetuning method should have three prerequisites: task compatibility, input compatibility, and evidence abundance. Bearing this belief in mind, this paper designs a novel prompt-based method (called DLM-SCS) for few-shot text classification, which utilizes the discriminative language model ELECTRA that is pretrained to distinguish whether a token is original or replaced. The method is built upon the intuitive idea that the prompt instantiated with the true label should have higher semantic consistency score than other prompts with false labels. Since a prompt usually consists of several components (or parts), its semantic consistency can be decomposed accordingly, which means each part can provide information for semantic consistency discrimination. The semantic consistency of each component is then computed by making use of the pretrained ELECTRA model, where no extra parameters get introduced. Extensive experiments have shown that our model outperforms several state-of-the-art prompt-based few-shot methods on 10 widely-used text classification tasks.

Keywords: discriminative language model, prompt

1. Introduction

Nowadays, with the upsurge of interest in a wide range of pretrained language models, the *pretraining-finetuning* paradigm (Radford et al., 2018; Dong et al., 2019) has become a de facto standard for various downstream NLU and NLG tasks. Different language models usually have different scopes of application. *Auto-regressive language models* (ARLM) such as GPT-3 (Brown et al., 2020) and Ernie-3 (Sun et al., 2021) predict the next token based on all the previous ones, usually in the left-to-right order. Since it is trained to encode a uni-directional context, it is not effective at downstream NLU tasks that often require bidirectional context information. In addition, these models are large and costly to finetune, or even not available publicly, which makes them impossible to use in the pretraining-finetuning paradigm. *Masked language models* (MLM) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) mask some tokens in inputs and are trained to reconstruct the original tokens based on their bidirectional surrounding context, which is often preferable in NLU tasks such as text classification but not applicable in NLG.

Conventional finetuning method for downstream text classification task usually builds up a classification head together with additional parameters on top of the special `[CLS]` token from scratch and fine-tunes the whole model. Such models work well with abundant training examples in rich data regimes, but will be cornered in the few-shot scenario, not to mention the zero-shot, because of the gap between pretraining and downstream tasks.

Initiated by the in-context learning of the GPT series (Radford et al., 2018, 2019; Brown et al., 2020), prompt-based method was first developed for zero-shot learning, and then studied by PET and iPET (Schick and Schütze, 2021a) for finetuning. After that, prompt-based learning methods have become increasingly popular, and have been proven to work effectively under few-shot or even zero-shot setting. To bridge the gap between the downstream task and the pretrained task, these methods transform downstream tasks into the same (or similar) form as the pretraining tasks solved during the original LM training with the help of textual prompts. Most existing prompt-based methods are using generative prompts that contain answer slots for various pretrained language models to fill in (Liu et al., 2021). As to the downstream text classification tasks, most works have been directed against pretrained masked language models (MLMs) by formulating downstream tasks as a masked language modeling task (Schick and Schütze, 2021a,b; Gao et al., 2021). A *template* converts the original input example x_{in} into a textual string (called *prompt*) $\tilde{x}$ that contains an unfilled `[MASK]` slot. A *verbalizer* is used to represent each class with a *label word* from the vocabulary. The model makes the prediction according to the probabilities of filling the `[MASK]` token with the label words. Such a prompt is called a *generative prompt*, which usually contains an an unfilled `[MASK]` as the answer slot, and the pretrained masked language model is finetuned to generate a correct label name to fill this answer slot. A simple prompt-based framework that treats MLM as masked token predictor for text classification is illustrated in Figure 1(b).

LREC-COLING 2024, pages 4968–4977
20-25 May, 2024.

Until the very recent, two prompt-based finetuning methods (Yao et al., 2022; Xia et al., 2022; Ni and Kao, 2022) have been proposed to exploit the pretrained ELECTRA (Clark et al., 2020) which is a *discriminative language model* (DLM). In contrast to the generative prompts, they use the prompts that contains no answer slot, which we call "*discriminative prompts*" and can be seen as the unmasked prompts which use label word(s) to fill in the `[MASK]` of generative prompts. The pretrained ELECTRA model is then applied on these discriminative prompts and tells us which label word is the original token (i.e., *not* a replaced token). However, these methods confine themselves only on the label word(s) in the discriminative prompts and expect the discriminative model to identify the semantic inconsistency incurred by the incorrect label words. This limited evidence is far from what can be obtained from the discriminative language model, and some available evidence is missing (Please refer to Section 3.2 for a simple motivating example).

The work done in this paper follows the thread of prompting the discriminative language model for few-shot text classification. The basic idea is that the DLM head can detect the discrepancy between inputs and label words. On the one hand, given an input example (a sentence or a sentence pair) and its true label, the DLM head is expected to assign low scores (or logits) to the salient tokens in the input example and the true label word. If a false label is given, it is desirable that the DLM head will assign high scores to both the false label word and the salient tokens in the input example.

To squeeze the most out of a pretrained language model such that it works best on a downstream few-shot learning task, three prerequisites are considered in designing a prompt-based method of finetuning a pretrained discriminative language model:

- **Prerequisite 1 (Task Compatibility):** As stated by most prompt tuning methods, the downstream task should be transformed into the same (or highly similar) form as the pretraining task, such that no (or few) additional parameters need introduced.
- **Prerequisite 2 (Input Compatibility):** The prompt template should be in the same form as the training data of the pretrained language model, such that the discriminative prompts are as much natural as possible. As a consequence, the pretrained language model can process them well and easily, without having to be tuned too far away.
- **Prerequisite 3 (Evidence/Information Abundance):** Last but not the least, the method should try its best to obtain and aggregate as much evidence and/or information as possible for decision making. Due to the nature of few-shot learning, it is unstable and has a big variance, and thus the aggregation of more evidence would be helpful in reducing the variance.

The contribution of this paper is threefold: (1) We propose a novel framework DLM-SCS[1] for few-shot text classification, which uses the pretrained discriminative language model ELECTRA as the semantic consistency scorer. (2) We design a method to measure the semantic consistency of a subsequence in the input prompt on the basis of the discriminative head of ELECTRA which can only measure the semantic inconsistency of each single token, and then use it to instantiate the framework into a concrete prompt-based finetuning model (also called DLM-SCS). (3) The proposed method has achieved the state-of-the-art performance on a variety of downstream sentence classification and sentence-pair classification tasks.

2. Related Work

2.1. Prompting MLM for Text Classification

Existing prompt-based learning methods for text classification usually reformulate the downstream text classification task into a cloze question task, and then finetune a pretrained masked language model to generate the most likely label word in the unfilled `[MASK]` position of the generative prompt (Schick and Schütze, 2021a). A lot of research effort has been devoted to the automatic construction of prompt templates and label words. (Schick et al., 2020) and (Schick and Schütze, 2021a) studied the automatic identification of label words. (Gao et al., 2021) made use of the pretrained seq2seq model T5 (Raffel et al., 2020) to generate template tokens in the template search process. Motivated by idea of in-context learning from GPT series (Brown et al., 2020), (Gao et al., 2021) used a single unmasked example prompt (called a *demonstration*) as additional context, which can boost the performance of prompt-based few-shot text classification task. (Park et al., 2022) made two extensions by multiple demonstrations and soft demonstration memory. In addition, (Zhang et al., 2021) proposed the DART method that optimizes the differentiable prompt template and label words by error backpropagation.

[1]Code available at https://github.com/liyahe/DLM-SCS.

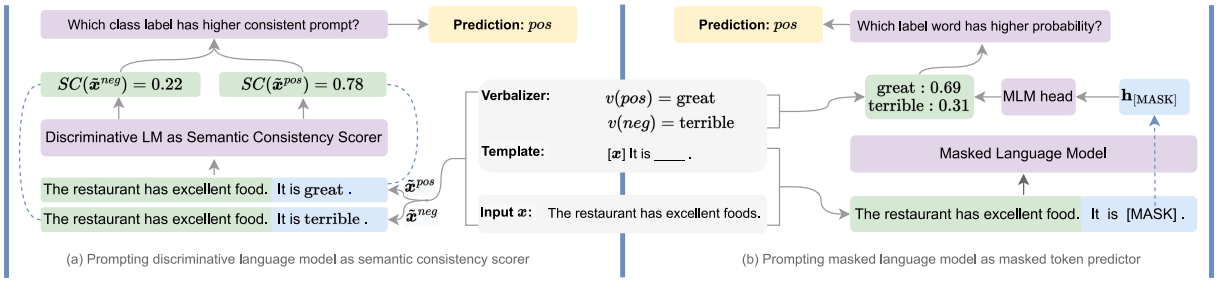

Figure 1: A schematic illustration of (a) our proposed DLM-SCS (Discriminative Language Model as Semantic Consistency Scorer), comparing to that of (b) traditional prompt-based model that uses masked language model as masked token predictor.

2.2. Prompting DLM for Text Classification

Two prompt-based text classification methods for finetuning discriminative language model (DLM) have been recently proposed (Yao et al., 2022; Xia et al., 2022; Ni and Kao, 2022). These methods reformulate text classification task into a discriminative language modeling problem, and predict the class label of an input example by using the DLM head to identify which label name is the original token instead of a replaced one. The DPT method proposed in (Yao et al., 2022) fills the input text $\boldsymbol{x}$ into the following template of discriminative prompt:

$$\texttt{[CLS]}\ \boldsymbol{x}\ \texttt{Class:}\ v(l_1), v(l_2), \ldots, v(l_n).\ \texttt{[SEP]}$$

where the verbalizer $v(\cdot)$ maps each class label l_i $(1 \leq i \leq n)$ to a distinct label word. Then the DLM head is used to judge which label word is proper in the context. It should be noted that DPT is not designed and also not suited for few-shot learning, because it does not satisfy the **Prerequisite 2** of *input compatibility*. As shown in Section 5, DPT cannot work well in few-shot scenario.

The other method, PromptELECTRA (Xia et al., 2022), was designed for few-shot text classification. Given an input example $\boldsymbol{x}$, it will generate one discriminative prompt for each possible class label. Thus, there are n discriminative prompts for $\boldsymbol{x}$. The DLM head is used to output the label word that has the highest probability of being original token in its corresponding prompt. This method satisfies **Prerequisite 1** and **2**, but it is not enough with respect to **Prerequisite 3** because it makes decision based on the only evidence from the candidate label words. Analogously, (Ni and Kao, 2022) presents that ELECTRA can also perform well on downstream tasks without fine-tuning.

Recently, prompting discriminative language models have been also applied to various tasks including medical text classification (Wang et al., 2023) and biomedical domain adaptation (Lu et al., 2023).

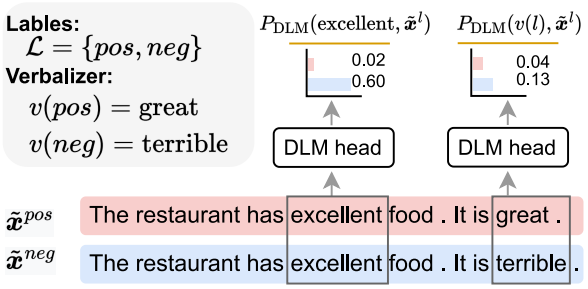

Figure 2: A simple example that motivates the DLM-SCS model.

3. Background and Motivation

3.1. The Pretrained ELECTRA Language Model

As a pretrained discriminative language model, ELECTRA (Clark et al., 2020) consists of an encoder and a discriminative head. The encoder first maps a sequence of input tokens $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ into a sequence of contextualized vector representations $[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$, and the discriminative head then predicts whether each token x_t $(1 \leq t \leq n)$ is a "real" or "replaced" token. In particular, the discriminative head simply applies a linear layer with the sigmoid activation function on the contextualized representation $\mathbf{h}_t$ of x_t:

$$P_{\text{DLM}}(x_t, \boldsymbol{x}) = \text{sigmoid}\left(\mathbf{w}^\top \mathbf{h}_t\right) \tag{1}$$

where $\mathbf{w}$ denotes the parameter vector. In this paper, we interpret the value of $\mathbf{w}^\top \mathbf{h}_t$ as the *unnormalized semantic inconsistency score* of token x_t in the context $\boldsymbol{x}$. The larger the value of $\mathbf{w}^\top \mathbf{h}_t$ is, the more semantically inconsistent the token x_t is, and the more likely the token x_t is a replaced token in $\boldsymbol{x}$.

3.2. Motivation

Figure 2 shows a simple input sentence from the sentiment classification task SST-2:

$\boldsymbol{x}$ = "The restaurant has excellent food."

The verbalizer v maps the class label pos (positive) to the label word "great", and the class label neg (negative) to "terrible". Therefore, two discriminative prompts ($\tilde{\boldsymbol{x}}^{neg}$ and $\tilde{\boldsymbol{x}}^{pos}$) are generated by using the template "$\boldsymbol{x}$ It is $v(l)$", one for each class label $l \in \mathcal{L} = \{pos, neg\}$.

Applying the discriminative head of the pretrained ELECTRA-large model to the label words: "terrible" in $\tilde{\boldsymbol{x}}^{neg}$ and "great" in $\tilde{\boldsymbol{x}}^{pos}$, we observe that $P_{\text{DLM}}(\text{terrible}, \tilde{\boldsymbol{x}}^{neg}) = 0.13$ and $P_{\text{DLM}}(\text{great}, \tilde{\boldsymbol{x}}^{pos}) = 0.04$. This evidence thus supports the conclusion that the input sentence is more likely to be positive, because the discriminative prompt $\tilde{\boldsymbol{x}}^{pos}$ of pos is semantically more consistent with respect to its label word (or in other words, $v(pos) =$ "great" is less likely to be a replaced token).

Besides the label tokens, the discriminative prompts contain tokens from the original input sentence, which may also provide us some evidence about the classification decision. As illustrated in Figure 2, by applying the DLM head on the token "excellent" in the discriminative prompts, we get $P_{\text{DLM}}(\text{excellent}, \tilde{\boldsymbol{x}}^{neg}) = 0.60$ and $P_{\text{DLM}}(\text{excellent}, \tilde{\boldsymbol{x}}^{pos}) = 0.02$, which also support the same conclusion that the input sentence is positive. This evidence from the token "excellent" is even stronger than the evidence provided by the label words.

4. Method

This section is devoted to a prompt-based framework for finetuning discriminative language models. The main thrust is to treat prompt-based text classification as a task of semantic consistency scoring, and calculate the semantic consistency of a prompt as a weighted average of the semantic consistency scores of multiple components (or parts) in the prompt.

Let $\mathcal{L}$ be the set of class labels for the target text classification task at hand. A verbalizer v is an injective function that maps each class label to a single token from M's vocabulary, $v : \mathcal{L} \to V$. We simply adopt the manual prompt templates used in the previous work (Gao et al., 2021). For sentence classification task, given an input example of single sentence $\boldsymbol{x}_{\text{in}} = \boldsymbol{x}^{(1)}$, we can generate a discriminative prompt $\tilde{\boldsymbol{x}}^l$ for each label $l \in \mathcal{L}$:

$$\tilde{\boldsymbol{x}}^l = \texttt{[CLS]}\ \boldsymbol{x}^{(1)}\ \texttt{It is}\ v(l)\ \texttt{.}\ \texttt{[SEP]} \quad (2)$$

For sentence-pair classification task, given an input example of sentence pair $\boldsymbol{x}_{\text{in}} = \left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right)$ and a label $l \in \mathcal{L}$, we can generate the discriminative prompt $\tilde{\boldsymbol{x}}^l$ as:

$$\tilde{\boldsymbol{x}}^l = \texttt{[CLS]}\ \boldsymbol{x}^{(1)}\ \texttt{?}\ v(l)\ \texttt{,}\ \boldsymbol{x}^{(2)}\ \texttt{[SEP]} \quad (3)$$

Therefore, there are $|\mathcal{L}|$ discriminative prompts $\{\tilde{\boldsymbol{x}}^l | l \in \mathcal{L}\}$ (one for each class label) generated for each input example.

The decision criterion of DLM-SCS model is based on the assumption that the discriminative prompt of true class label (*true prompt* in short) is semantically more consistent than the other discriminative prompts of false labels (*false prompts*). The class label l whose prompt $\tilde{\boldsymbol{x}}^l$ has the highest semantic consistency is chosen as the predicted label

$$\hat{l} = \arg\max_{l \in \mathcal{L}} SC(\tilde{\boldsymbol{x}}^l) \quad (4)$$

where $SC(\tilde{\boldsymbol{x}}^l)$ denotes the semantic consistency of the discriminative prompt $\tilde{\boldsymbol{x}}^l$. Figure 1(a) demonstrates this idea of using DLM as a semantic consistency scorer for text classification task.

Next, we move to the problem of how to calculate the semantic consistency $SC(\tilde{\boldsymbol{x}}^l)$ of a discriminative prompt $\tilde{\boldsymbol{x}}^l$. Since each prompt consists of several components (or parts), its semantic consistency can be decomposed accordingly. In particular, the semantic consistency of $\tilde{\boldsymbol{x}}^l$ is calculated as a weighted average of the semantic consistencies of some parts in $\tilde{\boldsymbol{x}}^l$:

$$SC(\tilde{\boldsymbol{x}}^l) = \lambda_0 \cdot sc\left(v(l), \tilde{\boldsymbol{x}}^l\right) + \sum_{\boldsymbol{x}^{(i)} \in \boldsymbol{x}_{\text{in}}} \lambda_i \cdot sc\left(\boldsymbol{x}^{(i)}, \tilde{\boldsymbol{x}}^l\right) \quad (5)$$

where:

- The term of form $sc(\boldsymbol{s}, \tilde{\boldsymbol{x}}^l)$ denotes the semantic consistency of a token subsequence $\boldsymbol{s}$ in the context of discriminative prompt $\tilde{\boldsymbol{x}}^l$. Here, $v(l)$ is seen as a subsequence of single token.
- The λ_i's are hyperparameters indicating the relative importance of the prompt parts ($i \in \{0, 1\}$ for sentence classification task, while $i \in \{0, 1, 2\}$ for sentence-pair classification).

Given a token subsequence $\boldsymbol{s}$ in the discriminative prompt $\tilde{\boldsymbol{x}}^l$, its semantic consistency can be simply measured by the softmax activation function over the mean negative logits of its tokens in the $|\mathcal{L}|$ discriminative prompts:

$$sc(\boldsymbol{s}, \tilde{\boldsymbol{x}}^l) = \frac{\exp\left(-\frac{1}{|\boldsymbol{s}|}\sum_{x \in \boldsymbol{s}} \mathbf{w}^\top \mathbf{h}_x^l\right)}{\sum_{l' \in \mathcal{L}} \exp\left(-\frac{1}{|\boldsymbol{s}|}\sum_{x \in \boldsymbol{s}} \mathbf{w}^\top \mathbf{h}_x^{l'}\right)} \quad (6)$$

where $\mathbf{h}_x^l$ denotes the contextualized representation of token x in the discriminative prompt $\tilde{\boldsymbol{x}}^l$. Here, because the value of $\mathbf{w}^\top \mathbf{h}_x^l$ denotes the semantic inconsistency score of x (i.e., the logit that x is a replaced token), the minus sign is introduced into the exponent part in order to transform it into

the score of semantic consistency. As to the semantic consistency of the label word $v(l)$ in $\tilde{\boldsymbol{x}}^l$, it is calculated with the following equation:

$$sc(v(l), \tilde{\boldsymbol{x}}^l) = \frac{\exp\left(-\mathbf{w}^\top \mathbf{h}^l_{v(l)}\right)}{\sum_{l' \in \mathcal{L}} \exp\left(-\mathbf{w}^\top \mathbf{h}^{l'}_{v(l')}\right)} \quad (7)$$

In Equation 6, all tokens in $\boldsymbol{s}$ are treated equally and their negative logits are simply averaged. However, different tokens should be of different importance with respect to the semantic consistency. Therefore, we make slight modification to Equation 6 by weighting each token with its inverse document frequency (IDF), as below:

$$sc(\boldsymbol{s}, \tilde{\boldsymbol{x}}^l) = \frac{\exp\left(-\frac{\sum_{x \in \boldsymbol{s}} \text{idf}(x) \mathbf{w}^\top \mathbf{h}^l_x}{\sum_{x \in \boldsymbol{s}} \text{idf}(x)}\right)}{\sum_{l' \in \mathcal{L}} \exp\left(-\frac{\sum_{x \in \boldsymbol{s}} \text{idf}(x) \mathbf{w}^\top \mathbf{h}^{l'}_x}{\sum_{x \in \boldsymbol{s}} \text{idf}(x)}\right)} \quad (8)$$

where $\text{idf}(x)$ of token x is set to be the inverse document frequency of the word that x belongs to. The inverse document frequency of words can be easily calculated from a large-scale unlabeled text corpus. In this paper, we simply calculate the token-level IDF values from unsupervised English Wikipedia corpus[2], and normalize the IDF values into the range of $[0, 1]$.

4.1. The Loss Function

Given a training example $\boldsymbol{x}_{\text{in}}$, each discriminative prompt $\tilde{\boldsymbol{x}}^l$ ($l \in \mathcal{L}$) contains m components/parts, where $m = 2$ for single-sentence classification, while $m = 3$ for sentence-pair classification. Each component/part corresponds to a subsequence of token in the prompt. The semantic consistency scorer of each prompt part actually outputs a probability distribution over the label set $\mathcal{L}$. Therefore, we use the cross-entropy function as the loss of each prompt part.

For the prompt part of label words, its loss is:

$$loss_0 = -\log sc(v(l^*), \tilde{\boldsymbol{x}}^{l^*}) \quad (9)$$

where l^* is the true class label of the input example $\boldsymbol{x}_{\text{in}}$.

For the part of a sentence $\boldsymbol{x}^{(i)}$ in the input example ($i = 1$ for single-sentence classification, while $i \in \{1, 2\}$ for sentence-pair classification), the loss is measured as:

$$loss_i = -\log sc(\boldsymbol{x}^{(i)}, \tilde{\boldsymbol{x}}^{l^*}) \quad (10)$$

Therefore, the total loss of the training example $\boldsymbol{x}_{\text{in}}$ is defined as the weighted average of the losses of its parts:

$$Loss = \sum_{0 \le i < m} \lambda_i \cdot loss_i \quad (11)$$

[2] https://dumps.wikimedia.org/enwiki

Task	Template	Label words
SNLI	<S_1>? $v(l)$, <S_2>	Yes/No/Maybe
MNLI	<S_1>? $v(l)$, <S_2>	Yes/No/Maybe
QNLI	<S_1>? $v(l)$, <S_2>	Yes/No
RTE	<S_1>? $v(l)$, <S_2>	Yes/No
MRPC	<S_1>? $v(l)$, <S_2>	Yes/No
QQP	<S_1>. $v(l)$, <S_2>	Yes/No
SST-2	<S_1>It is $v(l)$.	terrible/great
SST-5	<S_1>It is $v(l)$.	terrible/bad/ okay/good/great
MR	<S_1>It is $v(l)$.	terrible/great
CR	<S_1>It is $v(l)$.	terrible/great

Table 1: The Manual templates and label words used in the experiments.

where the hyperparameters λ_i's are the same as the ones used in Equation 5.

4.2. Model Optimization

For model training, we adopt AdamW algorithm and set a linear learning rate variation with warmup ratio of 0.05. For all the datasets, we take learning rate as 1e-5, and batch size as 4 for few-shot samples. For each trial, we train the model for 15 epochs, validate the performance every 50 steps, and take the best checkpoint. Early stopping is used to avoid overfitting.

As to the hyperparameters λ_i's, we set $\lambda_1 = 1 - \lambda_0$ for single-sentence classification, while $\lambda_1 = \lambda_2 = \frac{1-\lambda_0}{2}$ for sentence-pair classification. The best value of λ_0 is chosen based on its performance on the development set by a grid search from 0.0 to 1.0 with step size $\frac{1}{30}$.

5. Experimental Results

To evaluate the performance of our approach, we follow the experimental setting from (Gao et al., 2021). For each task, we take only K training examples per class for the training set $\mathcal{D}_{\text{train}}$, and thus the total number of training examples is $K \times |\mathcal{L}|$. A development set $\mathcal{D}_{\text{dev}}$ of the same size as the few-shot training set is employed for model selection and hyper-parameter tuning. Unless specified otherwise, the value of K is set to 16 by default, and the reported performance metrics are averaged over the same set of 5 random seeds.

We conduct extensive experiments on 4 sentence classification tasks (SST-2, SST-5, MR, and CR) and 6 sentence-pair classification tasks (SNLI, MNLI, QNLI, RTE, MRPC and QQP). The templates and label words used are provided in Table 1, which are the same as the ones used in (Gao et al., 2021). On these text classification tasks, our DLM-SCS method is compared with the

Model	SNLI (acc)	MNLI (acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)	SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)
Fine-tuning	48.4	45.8	60.2	54.4	76.6	60.7	81.4	43.9	76.9	75.8
	(4.8)	(6.4)	(6.5)	(3.9)	(2.5)	(4.3)	(3.8)	(2.0)	(5.9)	(3.2)
LM-BFF (man)	77.2	68.3	64.5	69.1	74.5	65.5	92.7	47.4	87.0	90.3
	(3.7)	(2.3)	(4.2)	(3.6)	(5.3)	(5.3)	(0.9)	(2.5)	(1.2)	(1.0)
+demonstrations	79.7	70.7	69.2	68.7	77.8	69.8	92.6	50.6	86.6	90.2
	(1.5)	(1.3)	(1.9)	(2.3)	(2.0)	(1.8)	(0.5)	(1.4)	(2.2)	(1.2)
LM-BFF (auto)	77.1	68.3	68.3	73.9	76.2	67.0	92.3	49.2	85.5	89.0
	(2.1)	(2.5)	(7.4)	(2.2)	(2.3)	(3.0)	(1.0)	(1.6)	(2.8)	(1.4)
+demonstrations	77.5	70.0	68.5	71.1	78.1	67.7	93.0	49.5	87.7	91.0
	(3.5)	(3.6)	(5.4)	(5.3)	(3.4)	(5.8)	(0.6)	(1.7)	(1.4)	(0.9)
DART	75.8	67.5	66.7	68.7	**78.3**	67.8	93.5	49.6	88.2	**91.8**
	(1.6)	(2.6)	(3.7)	(1.3)	(4.5)	(3.2)	(0.5)	(0.9)	(1.0)	(0.5)
DPT	47.4	39.0	54.6	50.2	76.4	56.1	92.6	44.0	89.5	91.2
	(7.7)	(1.8)	(5.4)	(2.8)	(6.1)	(1.1)	(1.3)	(3.8)	(2.1)	(1.6)
PromptELECTRA	79.1	65.8	70.9	68.2	73.5	63.1	93.1	51.4	89.4	90.2
	(3.4)	(2.5)	(2.1)	(2.8)	(4.6)	(3.3)	(1.0)	(2.2)	(1.6)	(1.4)
DLM-SCS (ours)	**82.2**	**71.0**	**77.0**	**75.0**	**78.3**	**72.2**	**93.6**	**51.5**	**90.2**	91.0
	(1.5)	(2.0)	(2.4)	(2.9)	(3.1)	(1.4)	(0.6)	(2.0)	(0.7)	(1.4)

Table 2: Performance evaluation on 6 sentence-pair classification tasks and 4 sentence classification tasks. The reported performance metrics are averaged over the same set of 5 random seeds, each random seed is used to sample 16 training example per class for training set, and the development set is of the same size as the training set. Both average results and standard deviations are reported above.

conventional fine-tuning method and several state-of-the-art prompt-based finetuning methods:

- **Fine-tuning**: The conventional fine-tuning of Roberta-Large in the few-shot experimental setting.
- **LM-BFF(man)**: The better few-shot fine-tuning of language models with manual prompts (Gao et al., 2021).
- **LM-BFF(auto)**: The better few-shot fine-tuning of language models with automatically searched templates (Gao et al., 2021). For both LM-BFF (man) and LM-BFF (auto), "+demonstrations" means incorporating demonstrations as additional context, which leads to performance gains in majority of tasks as indicated in (Gao et al., 2021).
- **DART**: The differentiable prompt framework proposed in (Zhang et al., 2022), where the prompt template and the target labels are differentially optimized with backpropagation.
- **DPT**: The prompt tuning framework for discriminative PLMs proposed in (Yao et al., 2022). Although DPT does not aim at few-shot learning, we also include it in our experimental comparison. Its original implementation[3] can only deal with sentence classification tasks, we make a straightforward extension by using the manual template: "`[CLS]` $\boldsymbol{x}^{(1)}$ $\boldsymbol{x}^{(2)}$ `Class:` $v(l_1)$, $v(l_2)$, ..., $v(l_n)$ `[SEP]`" for a sentence-pair classification task with n classes.
- **PromptELECTRA**[4]: The few-shot learning framework with discriminative pretrained models in (Xia et al., 2022).

5.1. Main Results

Table 2 reports the few-shot finetuning results of these methods on large-sized PLMs, where DLM-SCS, PromptELECTRA and DPT are based on the ELECTRA-large model, while all the other methods are based on the Roberta-large model. It can be easily observed that our DLM-SCS model is the best performer and has achieved the best performance on 9 of the 10 tasks among all the competitors. The Fine-tuning and DPT are not designed for the few-shot setting and have the worst performance. The conventional Fine-tuning method does not satisfy the **Prerequisite 1** of *Task Compatibility* because the finetuning task does not match the pretraining task and additional parameters get introduced, while DPT does not satisfy the **Prerequisite 2** of *Input Compatibility* because the prompts used are not natural. DLM-SCS outperforms PromptELECTRA on all the ten tasks, because DLM-SCS makes use of more evidence in the prompt than PromptELECTRA, which

[3] https://github.com/thunlp/DPT

[4] code available at https://github.com/facebookresearch/ELECTRA-Fewshot-Learning

Model	SNLI	MNLI	QNLI	RTE	MRPC	QQP	SST-2	SST-5	MR	CR
(full) DLM-SCS	**82.2**	**71.0**	**77.0**	**75.0**	**78.3**	**72.2**	93.6	**51.5**	90.2	**91.0**
-w.o. token weight	78.2	70.0	73.4	73.6	76.9	69.6	93.0	48.8	**90.3**	90.3
-only label word	76.5	64.6	69.0	71.8	74.8	64.2	**93.7**	51.1	88.8	90.4

Table 3: Ablation study. The row "w.o. token weight" removes the IDF weights from Equation 8 (i.e., use Equation 6 instead of Equation 8). The row "only label word" removes the semantic consistency of the input example, and use simply the label words to measure the semantic consistency of a discriminative prompt.

Model	MM-SP	MM-S
Fine-tuning	57.7	69.5
LM-BFF (man)	69.9	79.4
+demonstrations	72.7	80.0
LM-BFF (auto)	71.8	79.0
+demonstrations	72.2	80.3
DART	70.8	80.8
DPT	54.0	79.3
PromptELECTRA	70.1	81.0
DLM-SCS (ours)	**76.0**	**81.6**

Table 4: The mean performance metrics of compared methods. MM-SP denotes the mean performance metric on the 6 sentence-pair tasks, and MM-S denotes the mean performance metric on the 4 single-sentence tasks.

has manifested the value of **Prerequisite 3** (*Evidence/Information Abundance*).

To provide an intuitive understanding of the overall performance lifting extent achieved by our DLM-SCS method, Table 4 reports the mean performance metrics averaged on the sentence-pair tasks and the single-sentence tasks. It can be observed that:

- On the sentence-pair tasks, a substantial performance lifting is observed. Our DLM-SCS model achieves the highest mean metric score 76.0, superseding the score 72.7 of the runner-up LM-BFF(man)+demonstrations by a large margin.
- On the single-sentence tasks, the DLM-SCS model also achieves the highest mean metric 81.6, compared with 81.0 of the runner-up PromptELECTRA.

The extent of performance lifting on sentence-pair tasks is much larger than that on single-sentence tasks. One potential reason is that it aggregates three kinds of evidence (from the first sentence, the second sentence, and the label word) for sentence-pair tasks. In addition, "+demonstrations" can improve the performance of LM-BFF(man) and LM-BFF(auto), because the demonstrations provide more information in the context. As a conclusion, **Prerequisite 3** of *Evidence Abundance* is highly valuable in few-shot setting.

5.2. Ablation Study

Two main techniques play important roles in the DLM-SCS model: the first is to integrate the evidences from multiple components (or parts) of the prompt, and the second is to weight tokens in each prompt part with IDF values. Table 3 shows the effects on the model performance by removing each technique. It can be seen that either of the techniques is indispensable: the removal of any technique will lead to substantial reduction of model performance. The evidence from the input example part of the prompt plays a crucial role in the success of our model, and its integration with the evidence of label word leads to substantial performance boosting in classification accuracy, which is consistent with the Prerequisite 3 we proposed.

5.3. Varying the number of training examples

Figure 3 compares the performance DLM-SCS with two competitors LM-BFF (man) and PromptELECTRA as the number of training examples (K) increases from 16 to 256. It can be observed that:

- On 9 datasets (except MRPC), DLM-SCS consistently outperforms LM-BFF (man) when K varies from 16 to 256.
- On 7 datasets (except MRPC, SST-2 and SST-5), DLM-SCS consistently outperforms PromptELECTRA for all K values. On SST-2 and SST-5, DLM-SCS wins PromptELECTRA for most K values (4 out of 5).

5.4. Reject Option: Unanimous vs. Disagreed Examples

The DLM-SCS model for text classification can be thought of as a weighted average of m semantic consistency scorers (refer to Equation 5). For sentence classification task, there are two component scorers ($m = 2$), one for the label words and the other for the input sentence. For sentence-pair classification task, there are three component scorers ($m = 3$), one for the label words, one for the

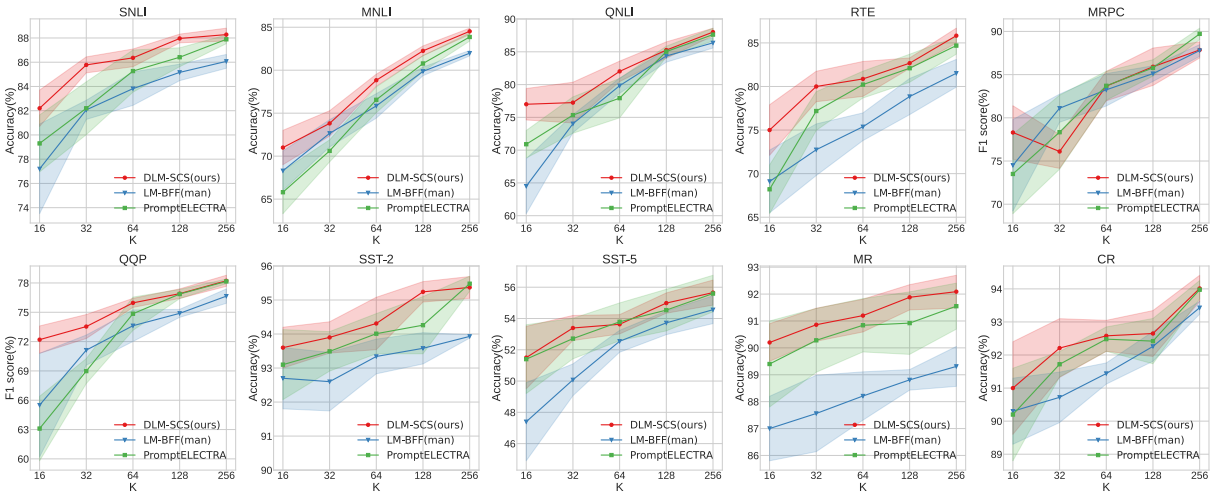

Figure 3: Performance comparison of DLM-SCS, LM-BFF (man) and PromptELECTRA with different numbers of training examples. The shadow area denotes the standard deviation of the performance metric.

Dataset	O.M	U.M	D.M	U.R
SNLI (Acc)	82.9	85.3	41.7	92.0%
MNLI (Acc)	71.1	75.4	41.6	82.0%
QNLI (Acc)	78.5	81.4	37.2	86.9%
RTE (Acc)	73.3	82.9	56.7	67.5%
MRPC (F1)	79.4	81.6	33.9	83.3%
QQP (F1)	71.4	75.1	28.0	87.5%
SST-2 (Acc)	93.3	94.8	30.0	99.0%
SST-5 (Acc)	53.9	56.4	44.1	74.3%
MR (Acc)	90.8	91.7	34.9	96.9%
CR (Acc)	91.3	93.5	59.3	93.3%

Table 5: Performance analysis of unanimous versus disagreed test examples. The column O.M denotes the overall performance metric on the whole test dataset, U.M denotes the performance on the unanimous test examples, D.M denotes the performance on the disagreed test examples, while the final U.R denotes the ratio of unanimous examples in the test dataset.

first sentence, and the other for the second sentence. Therefore, the test examples can be splitted into two parts: an example is called *unanimous* if all component scorers make the same decision for it; otherwise, it is a *disagreed* example,

Reject option means that a classifier can refuse to make a decision of a test example if the decision is thought of not sufficiently reliable, which is an important technique to improve the reliability of decision making. Traditional technique for reject option relies on whether the predictive probability (or predictive confidence) is higher than a pre-set threshold, which can be called *quantitative reject option*. However, in few-shot learning scenario, it is almost impossible to obtain reliable predictive probabilities. Instead, DLM-SCS can adopt the following *qualitative reject option* technique: it refuses to make prediction for a disagreed example.

From Table 5, it can be seen that the proportion of unanimous examples is high on the whole (larger than 80% on 8 datasets), which yields to the relatively low reject ratio. In addition, the performance metrics of disagreed examples are much lower than those of unanimous examples, which justifies the refusal of making decision for these disagreed examples.

6. Conclusion and Future Work

In this paper, we present DLM-SCS, a simple framework for finetuning discriminative language model using only a few examples, where the discriminative language model is used as a semantic consistency scorer of discriminative prompts. Given an input text example, it first constructs its discriminative prompts for all class labels, then calculates the consistency scores of these prompts, and finally outputs the class label whose prompt has the highest consistency. To calculate the consistency score of a prompt, it is decomposed into consistency scores of different prompt parts. The extensive empirical evaluation has shown that it achieves state-of-the-art performance on sentence (or sentence-pair) classification tasks.

At its current state, the DLM-SCS method simply works with manual and discrete prompt templates which may be suboptimal. It would be interesting to investigate how to adapt the techniques of automatically prompt generation (Gao et al., 2021) and/or differentiable prompting (Zhang et al., 2022) to our discriminative framework. In addition, it is also valuable to explore how to combine the discriminative PLMs and the generative PLMs, in order to achieve better performance.

7. Acknowledgements

This work is financially supported by National Natural Science Foundation of China (No.62076072). We are grateful to the anonymous reviewers for their valuable comments.

8. Bibliographical References

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*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

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

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*.

Keming Lu, Peter Potash, Xihui Lin, Yuwen Sun, Zihan Qian, Zheng Yuan, Tristan Naumann, Tianxi Cai, and Junwei Lu. 2023. Prompt discriminative language models for domain adaptation. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 247–258.

Shiwen Ni and Hung-Yu Kao. 2022. Electra is a zero-shot learner, too. *arXiv preprint arXiv:2207.08141*.

Eunhwan Park, Dong Hyeon Jeon, Seonhoon Kim, Inho Kang, and Seung-Hoon Na. 2022. LM-BFF-MS: improving few-shot fine-tuning of language models based on multiple soft demonstration memory. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 310–317. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification.

In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5569–5578. International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. 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.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. 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 2339–2352. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *CoRR*, abs/2107.02137.

Yu Wang, Yuan Wang, Zhenwan Peng, Feifan Zhang, Luyao Zhou, and Fei Yang. 2023. Medical text classification based on the discriminative pre-training model and prompt-tuning. *Digital Health*, 9:20552076231193213.

Mengzhou Xia, Mikel Artetxe, Jingfei Du, Danqi Chen, and Ves Stoyanov. 2022. Prompting ELECTRA: few-shot learning with discriminative pre-trained models. *CoRR*, abs/2205.15223.

Yuan Yao, Bowen Dong, Ao Zhang, Zhengyan Zhang, Ruobing Xie, Zhiyuan Liu, Leyu Lin, Maosong Sun, and Jianyong Wang. 2022. Prompt tuning for discriminative pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3468–3473. Association for Computational Linguistics.

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.

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*.