论文分享

Towards Unified Prompt Tuning for Few-shot Text Classification

from: EMNLP
research institutions: Alibaba Group

汇报人：张逸群
时间：2022.12.22

Towards Unified Prompt Tuning for Few-shot Text Classification

2022.12.21

Northeastern University

Abstract

Prompt-based fine-tuning has boosted the performance of Pre-trained Language Models(PLMs) on few-shot text classification by employing task-specific prompts. Yet, PLMs are unfamiliar with prompt-style expressions during pre-training, which limits the few-shot learning performance on downstream tasks. It would be desirable if the models can acquire some prompting knowledge before adaptation to specific NLP tasks. We present the *Uni*fied Prompt Tuning (UPT) framework, leading to better few-shot text classification for BERT style models by explicitly capturing prompting semantics from non-target NLP datasets. In *UPT*, a novel paradigm *Prompt-Options Verbalizer* is proposed for joint prompt learning across different NLP tasks, forcing PLMs to capture task-invariant prompting knowledge. We further design a self-supervised task named Knowledge-enhanced Selective Masked Language Modeling to improve the PLM's generalization abilities for accurate adaptation to previously unseen tasks. After multi-task learning across multiple tasks, the PLM can be better prompt-tuned towards any dissimilar target tasks in low-resourced settings. Experiments over a variety of NLP tasks show that *UPT* consistently outperforms state of-the-arts for prompt-based fine-tuning

基于提示的微调通过采用特定的任务提示，提高了预训练的语言模型（PLMs）在少量文本分类中的表现。然而，在预训练期间，PLMs不熟悉提示风格的表达方式，这限制了下游任务中的少量学习性能。如果模型能够在适应特定的NLP任务之前获得一些提示知识，那将是非常理想的。我们提出了统一提示调整（UPT）框架，通过明确捕捉来自非目标NLP数据集的提示语义，为BERT风格的模型带来更好的几页文本分类。在UPT中，我们提出了一个新颖的范式Prompt-Options Verbalizer，用于在不同的NLP任务中进行联合提示学习，迫使PLMs捕捉任务中的提示知识。我们进一步设计了一个自我监督的任务，名为 "知识增强的选择性屏蔽语言建模"，以提高PLM的泛化能力，准确适应以前未见过的任务。在对多个任务进行多任务学习后，PLM可以更好地对低资源环境中任何不同的目标任务进行提示调整。对各种NLP任务的实验表明，UPT在基于提示的微调方面一直优于现有技术。

All datasets are publicly available. Source codes will be released in EasyNLP (Wang et al., 2022). URL: https://github.com/alibaba/EasyNLP

Introduction

PLM的出现促进了许多NLP任务的发展。但是在fine-tuning中，由于模型的过拟合，PLM可能会因为训练样本少而表现不佳。

为了缓解低资源场景下的这个问题，NLP pormpt已经被广泛应用于实现zero-shot或者few-shot的PLM中。为了使prompt更加灵活和更加适应下有任务，prompt-tuning冻结了PLM的主干，并调整prompt。这种类型的方法特别适合难以微调的超大型PLM。对于Bert-style的PLM，prompt-tuning被提出，将文本分类任务转换成cloze类的任务。特定任务的离散模板，并在输入文本中加入mask标记。由MLM预测被掩码位置的标记来用于类别标签预测。因此，预训练模型获得的知识通过这种re-using的方式能够更好的用于下游任务。

最近有一些工作，专注于超大型PLM的多任务prompt tuning。具体来说，他们在完整训练样本上调整PLM来使PLM学习更多的提示知识，并且通过zero-shot学习对下游任务来做出预测。然而，我们观察到性能并不好，有以下两个原因：（1）PLM对不同的提示模板设计很敏感，不能适应有新提示的目标任务。（2）提示文本与PLM之间存在词汇分布差别。

我们想，如果Bert-style的PLM能够在适应下游任务之前获得提示知识，那就好了。因此，一个自然的问题出现了，如何才能使Bert-style模型在适应目标NLP任务前获得更多的提示知识?

为了解决这个问题，我们介绍了一个新框架叫做UPT（Unified Prompt Tuning），通过捕捉来自非目标数据集的一般提示语义，为Bert-style的PLM获得更好的zero-shot文本分类性能。具体来说，我们提出了一个统一的模板叫做prompt-options-verbalizer（POV），能够通过一系列不同类型的非目标NLP任务来混合prompt-tuning。为了进一步提高模型在未见过任务上的泛化能力，我们提出了一个新的辅助任务，名叫知识增强的选择性MLM。在多任务训练完成后，底层的PLM可以使用相同的提示模板来被微调成适应任何few-shot任务。

在实验中，我们验证了UPT在各种公开NLP数据集上的有效性。实验结果表明，UPT在基于提示的few-shot上优于其他方法，本文贡献如下：

- 我们介绍了新的UPT框架去改进基于提示的Bert-style模型。
- 在UPT中，一个新的模板POV被提出能够融合不同的NLP任务。为此我们设计了一个自监督KSMLM任务去改进PLM的能力。
- 在各种NLP数据集的广泛实验表明，UPT在基于提示的微调方面始终以相对较大的幅度优于最先进的技术。

UPT: *The Proposed Framework* *A Brief Overview of UPT*

$\mathcal{D}^*$是一个NLP任务$\mathcal{T}^*$的N-way-K-shot的训练集
PLM的参数是Θ
few-shot的学习目标是基于$\mathcal{D}^*$为任务$\mathcal{T}^*$获得一个高性能的模型，模型初始参数是Θ
$\mathcal{D}^*$的大小是$N \times K$，模型性能会受到很大的限制。
我们假定有M个其他NLP任务与$\mathcal{T}^*$不相似：$\mathcal{T}^{(1)}, \cdots, \mathcal{T}^{(M)}$，其训练集表示为$\mathcal{D}^{(1)}, \cdots, \mathcal{D}^{(M)}$（通常是非few-shot的）
UPT试图探索如何使用$\mathcal{D}^{(1)}, \cdots, \mathcal{D}^{(M)}$来增强PLM在$\mathcal{T}^*$的表现（基于训练集$\mathcal{D}^*$）

在UPT，模型首先在全资源任务$\mathcal{T}^{(1)}, \cdots, \mathcal{T}^{(M)}$下训练，目标是学习提示的语义以及通过提示来获得解决下游任务的方法。
然后，在具体的目标任务$\mathcal{T}^*$使用prompt-tuned进行训练。
为了统一学习过程，每一个训练样本i在不同的任务中都被使用相同的格式增强（通过Prompt-Options-Verbalizer（P_i, O_i, V_i），其中P_i是指prompt，O_i是指包含标签词的集合，V_i是目标词语到类别标签的映射。

此外，我们观察到标签的多样性在原始任务$\mathcal{T}^{(1)}, \cdots, \mathcal{T}^{(M)}$受到限制。对于以前没见过的任务，仅对这些任务进行优化可能会得到一个很差的泛化模型。因此我们进一步引入了一个自监督的知识增强性选择MLM（KSMLM）$\tilde{\mathcal{T}}$来作为一个辅助任务。

具体来说，取源任务的训练数据集中
$\tilde{\mathcal{D}} = \mathcal{D}^{(1)} \cup \mathcal{D}^{(2)} \cup \cdots \cup \mathcal{D}^{(M)}$的句子作为输入。句子被有选择地屏蔽，其选项由从大量语料库中挖掘的丰富知识生成。

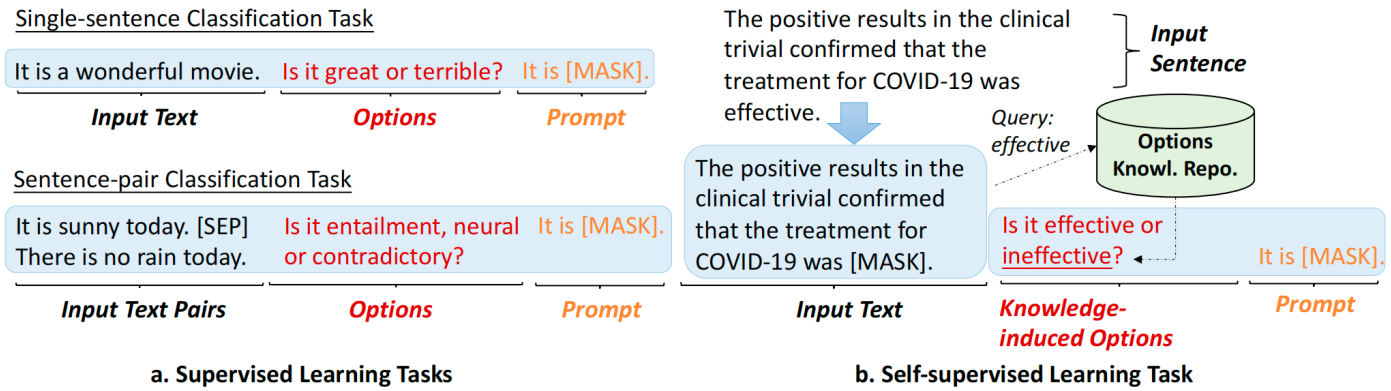


Figure 1: *UPT* is a unified framework that learns prompting knowledge from non-target NLP datasets to improve the performance on target tasks, in the format of *Prompt-Options-Verbalizer* (Sect. 2.2). Figures a) and b) show examples of supervised and self-supervised learning tasks (i.e., *Knowledge-enhanced Selective MLM*, Sect. 2.3).

The Unified Prompting Paradigm

一个基本的挑战是对于Bert-style的模型，不同nlp任务$\mathscr{D}^{(1)}, \cdots, \mathscr{D}^{(M)}$的标签集合的词汇是不同的。当处理混合样本的时候，一个简单的解决方案是建立一个统一的输出预测空间，包含所有的候选标签词。然而，扩大的输出空间让PLM难以优化。此外，输出预测空间可能不包括所有没见过的NLP任务的标签词。

在此，我们提出一个统一的提示范式，即通过提示-选项-映射三者来增加每个样本i的内容(P_i, O_i, V_i)，P_i是提供任务指导的提示，O_i是一个固定的式子，明确让模型在所有候选标签词中进行选择；为了便于快速适应任意的任务，V_i将被mask的语言标记输出映射到整个词汇表V。

在训练样本下的输出概率如下：

$$q(v \mid i, P_i, O_i, \Theta) = \frac{exp(s(v \mid i, P_i, O_i, \Theta))}{\sum_{v' \in \mathscr{V}} exp(s(v' \mid i, P_i, O_i, \Theta)))}$$

多任务提示loss可以写成如下公式：

$$\mathscr{L}_{MP} = -\sum_{i \in \mathscr{D}} P(\mathscr{V} \mid i, P_i, O_i, \Theta) \cdot log Q(\mathscr{V} \mid i, P_i, O_i, \Theta)$$

此外，我们注意到$\mathscr{D}^{(1)}, \cdots, \mathscr{D}^{(M)}$可以是具有不同大小任意标记的数据集，直接在其原始数据集上优化$\mathscr{L}_{MP}$可能会导致其偏向更大的数据集。为了解决这个问题，我们进行分层抽样，形成一个batch，其中一个训练样本随机抽取自$\mathscr{D}^{(1)}, \cdots, \mathscr{D}^{(M)}$，被选中的概率与数据集大小有关：

$$w_i = \frac{log \mid \mathscr{D}^{(k)} \mid + \gamma}{M \cdot \gamma + \sum_{k'=1}^{M} log \mid \mathscr{D}^{(k')} \mid}$$

因此，基于权重选择的多任务提示损失$\mathscr{L}_{WMP}$如下：

$$\mathscr{L}_{WMP} = -\sum_{i \in \mathscr{D}} w_i \cdot P(\mathscr{V} \mid i, P_i, O_i, \Theta) \cdot log Q(\mathscr{V} \mid i, P_i, O_i, \Theta)$$

Extending Unified Prompting to Self-supervised Learning

上述方法的缺点是，在这些监督学习任务中词汇表通常受限。在有新标签词的任务中，该模型不会被很好地推广。因此，我们利用了MLM预训练的想法。拟定了POV范式。

作为一个简单的方法，给定一个句子，我可以随机掩码一个词，并生成被掩码词的正确选项和一个随机单词，然后要求模型进行预测。但是，这种看似可行的方法会破坏训练过程，因为不是所有的词都是适合的标签词（例如，stopwords以及大量的动词和副词都没有在下游任务中被使用过）。选项中使用的替代词应该是合理的，才能使模型学到真正有用的知识。

为了解决这个问题，我们提出了自监督的KSMLM任务，见右图

P-Generation：这个过程的目的是为每个句子生成一个提示模板，每个句子都有一个[mask]标记。在多任务阶段，该模板被固定为"It is [MASK]".在特定任务阶段，使用自动生成模板(LM-BFF)。

O-Generation: 由于大多数用于语言理解的标签词都是形容词，因此，我们检测所有的形容词并过滤掉低频的形容词。这些形容词通过K-measn聚类。在形式上，我们构建了一个名为OKR的知识库。有了知识库的存在，我们可以生成由知识引起的选项。给定一个句子，生成选项"是[x1]还是[x2]？在知识库中选择最相似和最不相似的类中的词作为选项。

V-Generation：对于映射，我们将选项中的真实选项和生成的标签词映射到两个类（正确，不正确）。

Loss function：

$$\mathscr{L}_{KSMLM} = -\sum_{i\in\tilde{\mathscr{D}}} P(\mathscr{V}\,|\,i, P_i, O_i, \Theta)\cdot log Q(\mathscr{V}\,|\,i, P_i, O_i, \Theta)$$

$$\mathscr{L} = \mathscr{L}_{WMP} + \lambda\cdot\mathscr{L}_{KSMLM}$$

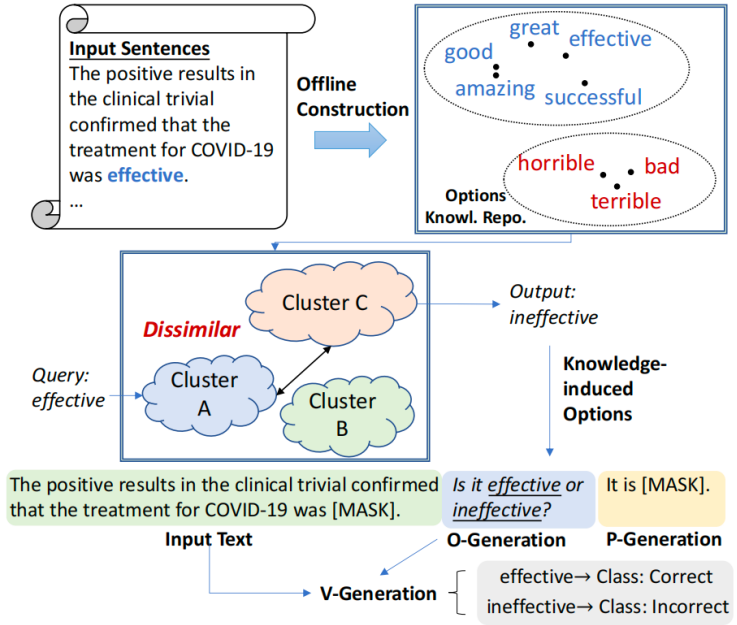


Figure 2: An illustrated example of the *POV* generation process for the *KSMLM* task.

Experimental Setting

在实验中，我们采用了9个公共文本分类数据集来评估UPT框架，这些数据集分为三组：情绪分析（SST-2，MR，CR），自然语言推理（MNLI，SNLI，QNLI，RTE）和相似性任务（MRPC、QQP）。

如上所述，在UPT期间，我们只利用来自不同任务的完整训练数据，然后在低资源环境下对模型进行prompt-tuning。举例来说，当目标任务是SST-2时，训练数据来自NLI任务相似性任务。底层的PLM模型时RoBERTa-large。基线模型包括标准微调，以及最近提出的四种few-shot学习算法：PET、LM-BFF、P-tuning以及PPT。为了与这些单任务基线模型进行公平的比较，我们方法的一个变体（UPT-Single）也是只基于POV进行微调而不使用其他数据集。

由于我们使用不同的数据集来训练模型，我们还包括两个多任务方法，名为MT（zero-shot）以及MT（few-shot）。我们也实现了UPT的zero-shot版本。此外，给定一个有监督的NLP任务，多数prompt可以手工制作。通过使用prompt增强训练样本，我们能够自动实现自集合（self-ensemble）学习。对于self- ensemble版本的UPT，我们使用五种不同的prompt。对于每个输入的sample，我们随机选择一个选项的表达式和一系列映射。

所有的结果都是以平均准确率和标准偏差来评价的（五个随机seed）。

实验设置：学习率1e-5，$\gamma = 0.0001$,$\lambda = 0.1$。

Main Results

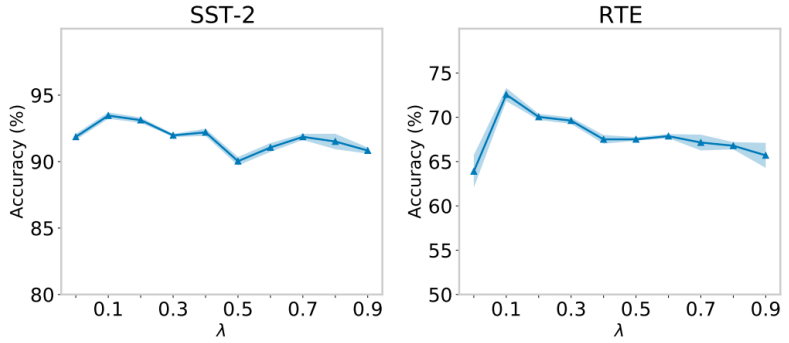
Paradigm	Method	Group 1: Sentiment. SST-2	MR	CR	Group 2: NLI. MNLI	SNLI	QNLI	RTE	Group 3: Paraphrase. MRPC	QQP	Avg.
Single-task methods w/o. the usage of dissimilar datasets ($K = 16$)											
FT	Fine-tuning	81.1 ±4.1	78.2±5.4	75.4±3.3	45.8±6.0	48.4±4.8	60.9±5.8	54.0±6.1	74.4±2.5	61.0±4.1	64.4±4.7
PT	PET	91.8±1.3	86.4±2.9	90.5±1.9	58.4±2.2	59.4±2.9	61.3±1.8	65.7±2.0	74.5±1.6	67.6±3.1	72.8±2.2
	LM-BFF	92.0±1.7	87.4±0.7	90.8±1.0	65.2±2.6	**71.7**±4.9	69.1±2.8	69.5±2.0	74.2±2.3	63.5±1.2	75.9±2.4
	P-Tuning	92.6±1.6	87.0±1.2	91.7±1.4	62.4±2.3	70.2±2.1	68.8±3.5	**70.8**±2.5	73.4±1.9	67.6±0.8	76.0±1.6
	PPT	92.3±0.5	87.1±1.6	90.9±1.3	64.9±2.0	71.4±1.5	68.8±2.9	67.9±2.6	74.8±2.1	67.2±1.2	76.1±1.8
	***UPT*-Single**	**92.9**±1.0	**87.7**±1.5	**91.8**±0.7	**65.6**±1.4	71.2±2.3	**70.1**±1.6	68.9±1.7	**75.1**±0.9	**72.1**±2.0	**77.2**±1.5
Multi-task methods w. the usage of dissimilar datasets ($K = 16$)											
PT	MT(Zero-shot)	58.7±1.6	59.0±3.6	58.9±2.8	36.3±3.3	39.2±3.2	40.9±2.5	54.9±1.4	70.6±2.6	42.8±2.5	51.3±2.2
	MT(Few-shot)	92.1±1.4	86.5±1.3	91.0±2.2	69.6±1.1	67.1±2.7	68.9±2.3	68.6±1.2	71.0±1.4	74.8±2.1	76.7±1.7
	UPT(Zero-shot)	74.5±1.2	73.9±1.3	72.4±1.4	43.7±2.0	46.0±2.1	53.9±1.9	57.1±1.0	70.7±0.9	56.5±1.3	61.0±1.5
	UPT	**93.5**±0.6	88.1±0.9	91.4±1.2	70.1±1.4	68.2±1.2	69.9±1.5	73.5±1.5	**77.0**±1.1	78.8±1.7	78.9±1.4
	***UPT*-SE**	93.1±0.4	**88.4**±0.9	**92.1**±1.0	**71.4**±1.1	**73.6**±0.6	**70.5**±1.6	**75.8**±0.8	76.2±0.4	**79.6**±1.3	**80.1**±1.1

Table 1: Comparison between *UPT* and baselines over all testing sets in terms of accuracy (%) and standard deviation. "FT" and "PT" refer to the *fine-tuning* and *prompt-based fine-tuning* paradigm, respectively. The methods in bold refer to our approach and its variants. The scores of baselines are re-produced using their open-source codes.

Model Analysis

参数分析

研究平衡系数λ的最佳选择。SST-2和RTE的结果显示在图3中。当λ=0.1时，我们有最好的性能。
λ=0.1，这表明我们提出的UPT具备泛化能力，当它被联合训练
在自监督的KSMLM任务上进行联合训练时，我们的UPT具有通用性。我们还观察到，当λ变大时，性能会下降。这意味着KSMLM是一个合适的正则化任务，但也可能引入大量的提示和与下游任务无关的选项。这为模型的改进提供了新的机会。



消融实验

为了清楚地验证UPT中每个成分的贡献，我们对所有组别进行了消融研究，并报告了每个组别平均数。对所有组别进行消融研究，并报告其平均 准确性。如表3所示
w/o POV：不使用任何options
w/o KSMLM：λ=0，与UPT-single相同
w/o OKR：随机选择替换标签词
w/o POV & KSMLM：不使用任何选项和辅助KSMLM任务

无论删除哪个模块，模型的性能移除哪个模块，模型的性能都会受到影响。特别是，当我们同时删除POV和KSMLM时，其性能分别下降了分别下降了1.4%、1.5%和4.4%。这个设置的准确度这个设置的准确率比没有设置时要低。POV和KSMLM下降的最多，这表明这两个组件对我们框架的高性能有很大贡献。

Method/Group	Group 1	Group 2	Group 3
MT (Few-shot)	89.9	68.6	72.9
UPT	**91.0**	**70.2**	**77.9**
w/o. *POV*	90.2	68.9	74.2
w/o. *KSMLM*	90.9	69.1	73.7
w/o. *POV&KSMLM*	89.6	68.7	73.5
w/o. *OKR*	90.7	69.9	76.8

Table 3: Ablation study in terms of accuracy (%). Standard deviations are omitted here to save space.

Model Analysis

Sample Efficiency

我们进一步探讨模型 的效果，每类训练样本的数量从16到512个不等。每类(K)的不同数量，从16到512。我们还使用标准 微调作为参考。如图4所示。每一点都是指5个随机抽样的数据集的平均得分。我们观察到，我们的UPT 始终取得了更高的分数，无论 训练样本的数量。此外，UPT的方差低于UPT的方差比微调低，这意味着我们方法的稳定性更好。这意味着我们的方法的稳定性更好。这一点与其他基于提示的方法不同。

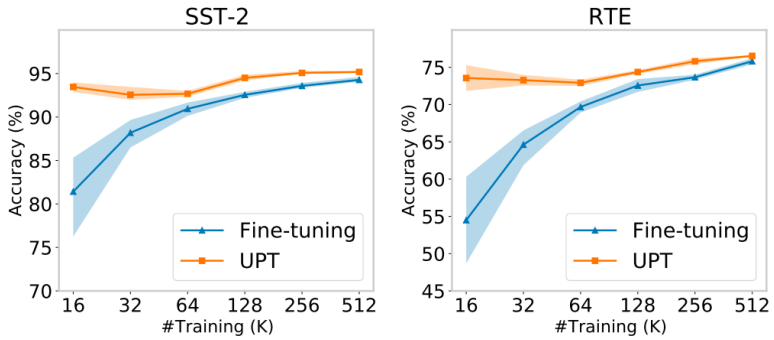


Figure 4: Results of sample efficiency analysis. We compare *UPT* with standard *fine-tuning* with different numbers of training samples K over two tasks.

Model Scale Analysis

为了进一步证明UPT可以提高模型的性能，而不考虑规模，我们把多个小规模的BERT视为模型骨干。我们在表2中只说明了SST-2、MR和CR的结果。我们还测试了不使用不同的NLP数据集，并显示了相对的改进。结果表明，模型的规模在模型泛化的能力中起着重要作用。我们还发现，使用不同数据集的UPT可以高度提高有效性，特别是在小规模的PLM上。因此，我们的方法更适合于生产高性能的在线应用的小型PLMs。

BERT Scale	SST-2	MR	CR	Avg.
Base	82.6+3.8	71.1+9.3	78.1+8.9	77.2+7.3
Medium	68.0+3.0	63.4+4.2	70.2+6.1	67.2+4.4
Small	66.3+3.7	58.1+4.6	68.2+5.5	64.2+4.6
Mini	58.8+3.1	59.4+7.6	65.8+7.5	61.3+6.1
Tiny	54.2+3.8	54.0+1.3	54.4+5.2	54.2+3.4

Table 2: Results of model scale analysis. We report the accuracy (%) of *UPT* based on BERT with other scales, and relative improvements, compared to the models w/o. prompt learning over *dissimilar* datasets.

Model Analysis

Adaptation Efficiency of Task Groups

我们专注于多任务训练，然后在低资源环境下对目标任务进行提示。因此，值得探讨的是，哪些/多少组任务对适应性的提高有更好的效果。具体来说，当给定一个目标任务（例如，MNLI），我们只选择一组的任务（如第3组的MRPC和QQP）进行多任务提示调优，然后在目标任务上对模型进行微调。如图5中，第i行和第j列的单元格表示从单任务学习到第j个任务的相对改进对第j个任务的学习的相对改进。第i组被添加到多任务提示学习中。为了直观起见，我们将每一列的值归一化，以表示在多任务提示学习中，单任务学习对第j个任务的影响百分比。

具体来说，以NLI为源组，我们从该组中随机选择M个数据集作为我们的源任务，然后在每个目标任务上对模型进行提示调整。然后对每个目标任务进行提示调整。图6的结果表明，当我们增加M值时，准确率会进一步提高。我们还发现，与MRPC和QQP相比，NLI的改进更为明显。我们认为，NLI更容易被适应于 的任务，因为它们都对句子对之间的关系进行了建模。句子对之间的关系。

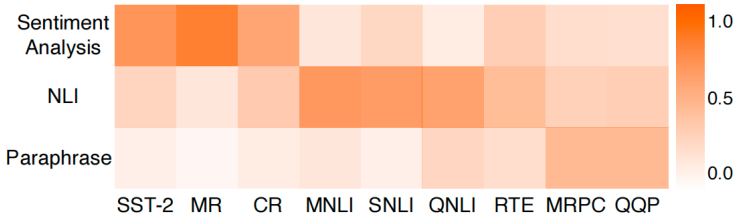


Figure 5: Adaptation efficiency between task groups. The shade of color indicates the degree of adaptation.

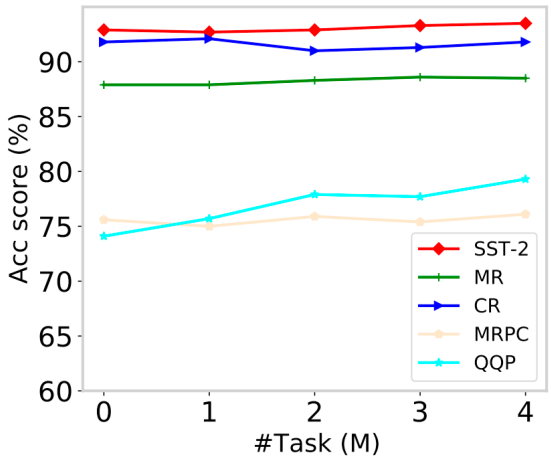


Figure 6: Adaptation efficiency between the different numbers of NLI tasks (M) and each target task from Sentiment and Paraphrase.

组会汇报

汇报人：张逸群
时间：2022.12.22

Model Analysis

Adaptation Efficiency of Task Groups

我们专注于多任务训练，然后在低资源环境下对目标任务进行提示。因此，值得探讨的是，哪些/多少组任务对适应性的提高有更好的效果。具体来说，当给定一个目标任务（例如，MNLI），我们只选择一组的任务（如第3组的MRPC和QQP）进行多任务提示调优，然后在目标任务上对模型进行微调。如图5中，第i行和第j列的单元格表示从单任务学习到第j个任务的相对改进对第j个任务的学习的相对改进。第i组被添加到多任务提示学习中。为了直观起见，我们将每一列的值归一化，以表示在多任务提示学习中，单任务学习对第j个任务的影响百分比。

具体来说，以NLI为源组，我们从该组中随机选择M个数据集作为我们的源任务，然后在每个目标任务上对模型进行提示调整。然后对每个目标任务进行提示调整。图6的结果表明，当我们增加M值时，准确率会进一步提高。我们还发现，与MRPC和QQP相比，NLI的改进更为明显。我们认为，NLI更容易被适应于 的任务，因为它们都对句子对之间的关系进行了建模。句子对之间的关系。

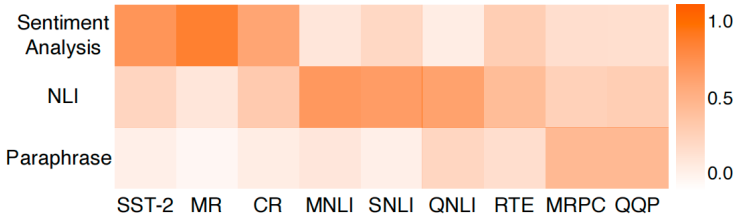


Figure 5: Adaptation efficiency between task groups. The shade of color indicates the degree of adaptation.

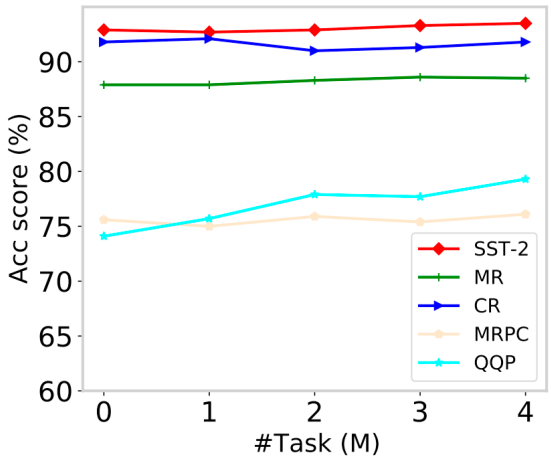


Figure 6: Adaptation efficiency between the different numbers of NLI tasks (M) and each target task from Sentiment and Paraphrase.