



黑马程序员线上品牌

# Prompt-Tuning方法入门

---

一样的教育，不一样的品质



# 目录

Contents

1. 什么是NLP四范式
2. Fine-Tuning回顾
3. Prompt-Tuning技术介绍
4. Prompt-Tuning入门方法

01

# NLP任务四种范式

## I NLP任务四种范式

目前学术界一般将NLP任务的发展分为四个阶段，即NLP四范式：

### 第一范式

基于「传统机器学习模型」的范式，如TF-IDF特征+朴素贝叶斯等机器算法。



### 第二范式

基于「深度学习模型」的范式，如word2vec特征+LSTM等深度学习算法，相比于第一范式，模型准确有所提高，特征工程的工作也有所减少。



### 第三范式

基于「预训练模型+fine-tuning」的范式，如Bert+fine-tuning的NLP任务，相比于第二范式，模型准确度显著提高，模型也随之变得更大，但小数据集就可训练出好模型



### 第四范式

基于「预训练模型+Prompt+预测」的范式，如Bert+Prompt的范式相比于第三范式，模型训练所需的训练数据显著减少



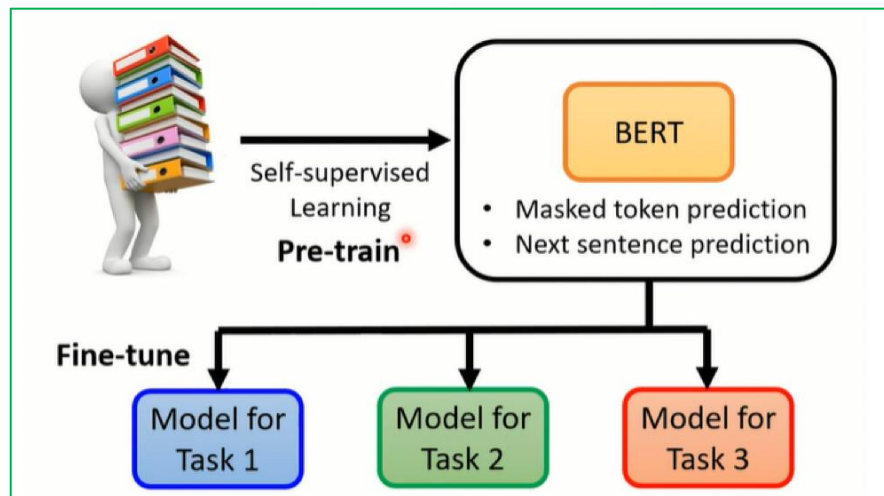
在整个NLP领域，整个发展历程是朝着精度更高、少监督，甚至无监督的方向发展的。而 Prompt-Tuning是目前学术界向这个方向进军最新也是最火的研究成果。

02

# Fine-Tuning回顾

## Fine-Tuning

Fine-Tuning属于一种迁移学习方式，在自然语言处理（NLP）中，Fine-Tuning是用于将预训练的语言模型适应于特定任务或领域。Fine-Tuning的基本思想是采用已经在大量文本上进行训练的预训练语言模型，然后在小规模的任务特定文本上继续训练它。



### 痛点

- 下游任务的目标和预训练的目标差距过大，可能导致过拟合。
- 微调过程中需要依赖大量的监督语料等等。

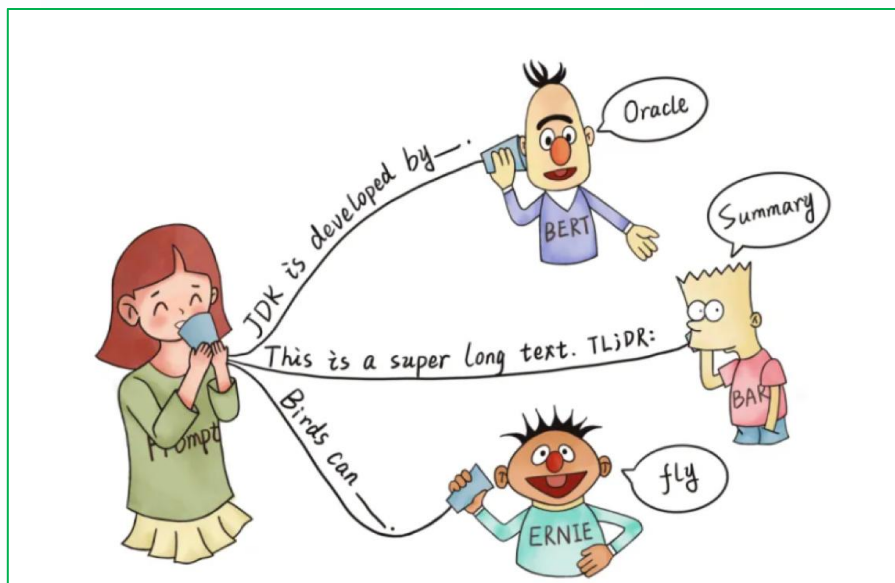
解决方法：Prompt-Tuning, 通过添加模板的方法来避免引入额外的参数，从而让模型可以在小样本（few-shot）或者零样本（zero-shot）场景下达到理想的效果。

# 03

## Prompt-Tuning技术发展介绍

## Prompt-Tuning

基于Fine-Tuning的方法是让预训练模型去迁就下游任务，而基于Prompt-Tuning的方法可以让下游任务去迁就预训练模型，其目的是将Fine-tuning的下游任务目标转换为Pre-training的任务。



那么具体如何工作呢？我们接下来将以一个二分类的情感分析为例子，进行简单理解：



## ■ Prompt-Tuning

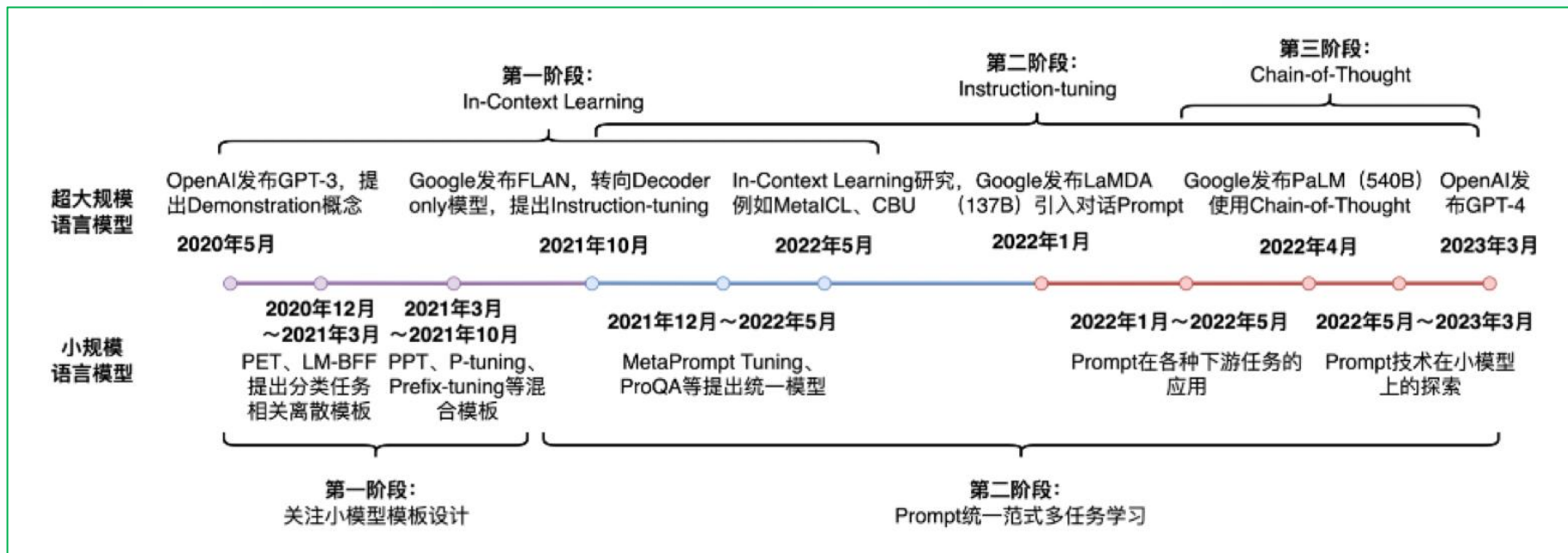
- 定一个句子: [CLS] I like the Disney films very much. [SEP]
- 传统的Fine-tuning方法: 将其通过BERT模型获得 [CLS] 表征之后再喂入新增加的MLP分类器进行二分类, 预测该句子是积极的 (positive) 还是消极的 (negative), 因此需要一定量的训练数据来训练.

### Prompt-tuning执行步骤:

- **构建模板** (Template): 生成与给定句子相关的一个含有[MASK]标记的模板. 例如It was [MASK], 并拼接到原始的文本中, 获得Prompt-Tuning的输入: [CLS] I like the Disney films very much. [SEP] It was [MASK]. [SEP]. 将其喂入BERT模型中, 并复用预训练好的MLM分类器, 即可直接得到[MASK]预测的各个token的概率分布.
- **标签词映射** (Verbalizer): 因为[MASK]只对部分词感兴趣, 因此需要建立一个映射关系. 例如如果[MASK]预测的词是“great”, 则认为是positive类, 如果是“terrible”, 则认为是negative类.
- **训练**: 根据Verbalizer, 则可以获得指定label word的预测概率分布, 并采用交叉信息熵进行训练. 此时因为只对预训练好的MLM head进行微调, 所以避免了过拟合问题

## Prompt-Tuning

- 不同的句子应该有不同的template和label word，因此如何最大化的寻找当前任务更加合适的template和label word是Prompt-tuning非常重要的挑战。



Prompt-tuning技术发展历程



# 思考总结

Thinking summary

## 1.NLP任务四范式？

答案：1.传统机器学习； 2.深度学习模型； 3.预训练+fine-tuning; 4.预训练+prompt+预测

## 2. 什么是Fine-Tuning？

答案：采用已经在大量文本上进行训练的预训练语言模型，然后在小规模的任务特定文本上继续训练它。

## 3. Prompt-Tuning的实现？

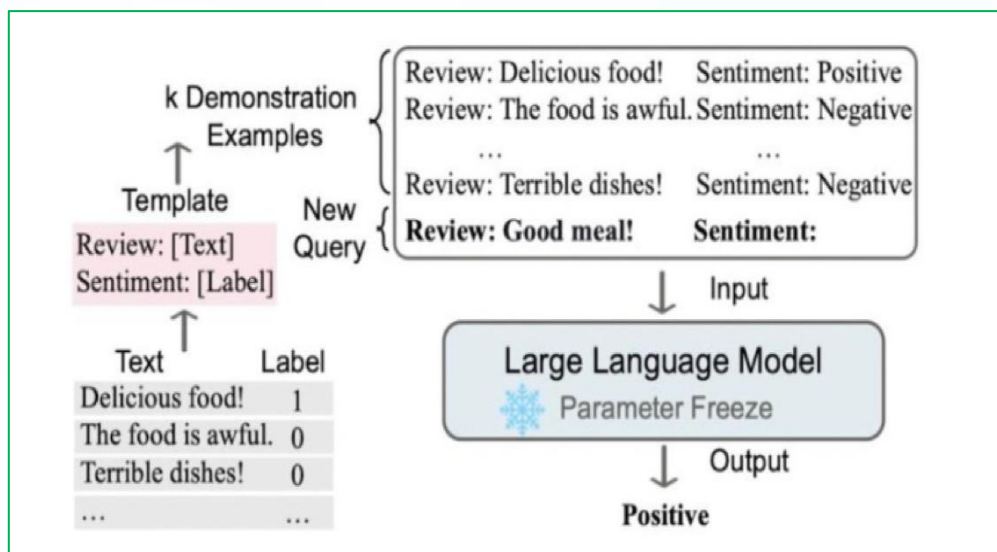
答案：1.构建模版； 2.标签词映射； 3.训练

# 03

## Prompt-Tuning主要方法

## ■ Prompt-Tuning的鼻祖-----GPT3

GPT-3开创性的提出了In-context Learning的思想. 即无须修改模型即可实现few-shot、zero-shot的 learning. 同时引入了Demonstrate Learning, 即让模型知道与标签相似的语义描述, 提升推理能力.



### 问题

- 建立在超大规模的预训练语言模型上.模型参数数量通常超过100亿, 在真实场景中很难应用.
- 该方法应用于参数规模小的模型, 效果会下降很多, 因此后续提出Prompt-Tuning.
- GPT3中提供的prompt过于简单, 泛化性能低.

## I PET模型

PET (Pattern-Exploiting Training) 出自《Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference》(EACL2021)，根据论文题目则可以看出，Prompt-Tuning启发于文本分类任务，并且试图将所有的分类任务转换为与MLM一致的完形填空。

PET模型提出两个很重要的组件(简称PVP):

➤ Pattern (Template) : 记作T, 即上文提到的Template, 其为额外添加的带有[mask]标记的短文本, 通常一个样本只有一个Pattern, 由于不同的任务、不同的样本可能会有其更加合适的pattern, 因此如何构建合适的pattern是Prompt-Tuning的研究点之一.

➤ Verbalizer: 记作V, 即标签词的映射, 对于具体的分类任务, 需要选择指定的标签词 (label word) .例如情感分析中, 我们期望Verbalizer可能是 (positive和negative是类标签) . 同样, 不同的任务有其相应的label word, 但需要注意的是, Verbalizer的构建需要取决于对应的Pattern, 因此如何构建Verbalizer是另一个研究挑战.

## PET模型

基于Pattern-Verbalizer-Pair (PVP) 组件训练目标:

$$p(y|x) = \prod_{j=1}^n p([mask]_j = \mathcal{V}(y) | \mathcal{T}(x))$$

目前基于PVP框架, 最需要关注的问题是**如何选择或构建合适的Pattern和Verbalizer**. 一种简单的方法是根据特定任务的性质和先验知识人工设计模板. 注意: 在同样的数据集和训练条件下. 选择不同的Pattern和Verbalizer会产生差异很大的结果.

Template	Label words	Accuracy
SST-2 (positive/negative)		mean (std)
$\langle S_1 \rangle$ It was [MASK] .	great/terrible	92.7 (0.9)
$\langle S_1 \rangle$ It was [MASK] .	good/bad	92.5 (1.0)
$\langle S_1 \rangle$ It was [MASK] .	cat/dog	91.5 (1.4)
$\langle S_1 \rangle$ It was [MASK] .	dog/cat	86.2 (5.4)
$\langle S_1 \rangle$ It was [MASK] .	terrible/great	83.2 (6.9)
Fine-tuning	-	81.4 (3.8)
SNLI (entailment/neutral/contradiction)		mean (std)
$\langle S_1 \rangle$ ? [MASK] , $\langle S_2 \rangle$	Yes/Maybe/No	77.2 (3.7)
$\langle S_1 \rangle$ . [MASK] , $\langle S_2 \rangle$	Yes/Maybe/No	76.2 (3.3)
$\langle S_1 \rangle$ ? [MASK] $\langle S_2 \rangle$	Yes/Maybe/No	74.9 (3.0)
$\langle S_1 \rangle$ $\langle S_2 \rangle$ [MASK]	Yes/Maybe/No	65.8 (2.4)
$\langle S_2 \rangle$ ? [MASK] , $\langle S_1 \rangle$	Yes/Maybe/No	62.9 (4.1)
$\langle S_1 \rangle$ ? [MASK] , $\langle S_2 \rangle$	Maybe/No/Yes	60.6 (4.8)
Fine-tuning	-	48.4 (4.8)



## I PET模型



因此如何能够自动地挑选合适的**PVP**? 本次课程主要着重介绍构建Pattern的方法: Prompt-Tuning. 接下来我们根据使用场景的不同, 分别介绍几种成熟的Prompt-Tuning方法.



## ■ Prompt-Oriented Fine-Tuning

Prompt-Oriented Fine-Tuning训练方法的本质是将目标任务转换为适应预训练模型的预训练任务，以适应预训练模型的学习体系。

情感分析任务为例：

### BERT Fine-Tuning

Fine-Tuning流程：将训练文本经过Bert编码后，生成向量表征，再利用该向量表征，连接全连接层，实现最终的情感类别识别，这种方式存在一个显式的弊端：预训练任务与下游任务存在gap.

VS

### Prompt-Oriented Fine-Tuning

基本流程：构建prompt文本：It was [MASK].，将prompt文本与输入text文本text = The film is attractive.拼接生成：It was [MASK].The film is attractive.，输入至预训练模型中，训练任务目标和MLM任务的目标一致，即识别被[MASK]掉的词。

## Prompt-Oriented Fine-Tuning

Prompt-Oriented Fine-Tuning方法中，预训练模型参数是可变的，本质是Prompt-Tuning+Fine-Tuning的结合体。该方法在Bert类相对较小的模型上表现较好，但是随着模型越来越大，如果每次针对下游任务，都需要更新预训练模型的参数，资源成本及时间成本都会很高，因此后续陆续提出了不更新预训练模型参数，**单纯只针对prompt进行调优的方法**。

针对Prompt调优方法的分类：**Hard Prompt** 和 **Soft Prompt**

常见下游任务的Prompt设计：

任务特性	模板
电影评论情感分析（二分类）	The movie review is [x]. It was [mask].
新闻分类	A [mask] News: [x].
实体识别	Shanghai is in the west of China. The entity Shanghai is [mask].
多项选择	Question: [q], Options: A. [x <sub>1</sub> ], B. [x <sub>1</sub> ], C. [x <sub>1</sub> ], D. [x <sub>1</sub> ]. The result is [mask].
文本摘要	Context: [x]. Abstract: [mask] [mask] [mask] ...

## I 模板类别

### Hard Prompt

离散提示: 是一种固定的提示模板, 通过将特定的关键词或短语(真实的文本字符串)直接嵌入到文本中, 引导模型生成符合要求的文本.

特点: 提示模板是固定的, 不能根据不同的任务和需求进行调整.

缺陷: 依赖人工, 改变prompt中的单个单词会给实验结果带来巨大的差异.

### Soft Prompt

连续提示: 是指通过给模型输入一个可参数化的提示模板, 从而引导模型生成符合特定要求的文本.

特点: 提示模板中的参数可以根据具体任务和需求进行调整, 以达到最佳的生成效果.

优点: 不需要显式地指定这些模板中各个token具体是什么, 而只需要在语义空间中表示一个向量即可.

接下来, 我们主要针对**Soft Prompt**方式进行讲述:

## I Soft Prompt理解

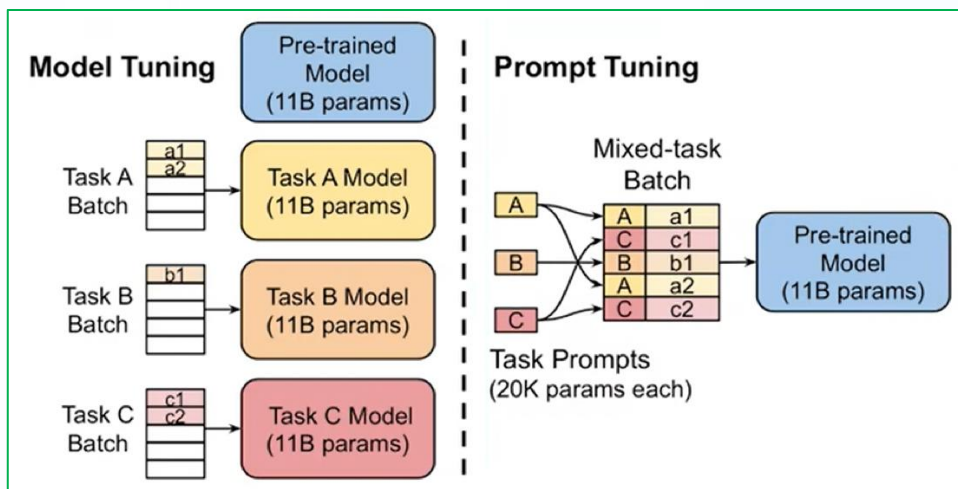
- 基于Soft Prompt, 不同的任务、数据可以自适应地在语义空间中寻找若干合适的向量, 来代表模板中的每一个词, 相较于显式的token, 这类token称为 **伪标记 (Pseudo Token)**. 下面给出基于连续提示的模板定义:
- 假设针对分类任务, 给定一个输入句子 $x$ , 连续提示的模板可以定义为 $T=[x],[v_1],[v_2],\dots,[v_n][\text{MASK}]$ : 其中 $[v_n]$ 则是伪标记, 其仅代表一个抽象的token, 并没有实际的含义, 本质上是一个向量.

总结来说: Soft Prompt方法, 是将模板变为可训练的参数, 不同的样本可以在连续的向量空间中寻找合适的伪标记, 同时也增加模型的泛化能力. 因此, 连续法需要引入少量的参数并在训练时进行参数更新, 但预训练模型参数是不变的, 变的是prompt token对应的词向量 (Word Embedding) 表征及其他引入的少量参数.

接下来, 我们将基于**Prompt-Tuning**讲解三种典型方法:

## Prompt Tuning方法 (NLG任务)

Prompt Tuning 是2021年谷歌在论文《The Power of Scale for Parameter-Efficient Prompt Tuning》中提出的微调方法，该方法基于T5模型(最大参数11B)为每一个输入文本假设一个固定前缀提示，该提示由神经网络参数化，并在下游任务微调时进行更新，整个过程中预训练的大模型参数被冻结。



思想

- ① 给定  $n$  个tokens, 记作  $x_1, \dots, x_n$ , 通过一个预训练模型对应的embedding table, 可以将  $n$  个token表示为一个向量矩阵 ( $X_e \rightarrow R^{n \times e}$ ).
- ② 将连续模板中的每个伪标记  $v_i$  视为参数, 可以通过另一个embedding table获得  $p$  个伪token标记为向量矩阵 ( $P_e \rightarrow R^{p \times e}$ ).
- ③ 将文本和Prompt拼接获得新的输入  $[P_e; X_e] \rightarrow R^{(p+n) \times e}$ .
- ④ 新的输入喂入一个MLP获得新的表征. 注意, 只有prompt对应的向量表征参数 ( $P_e \rightarrow R^{p \times e}$ ) 会随着训练进行更新.

## Prompt Tuning方法的特点

### 优点

- 大模型的微调新范式.
- 模型参数规模大了之后, 可以将大模型参数固定, 指定附加参数来适配下游任务, 而且适配性能基本和全参数微调相当.

### 缺点

- 在小样本学习场景上表现不太行.
- 收敛速度比较慢.
- 调参比较复杂.

## P-Tuning V1方法 (NLU任务)

P-Tuning 是2022年清华在论文《GPT Understands, Too》中提出的微调方法，P-Tuning V1方法的提出主要是为了解决这样一个问题：大模型的 Prompt 构造方式严重影响下游任务的效果。

Prompt	P@1
[X] is located in [Y]. ( <i>original</i> )	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08

解决方法



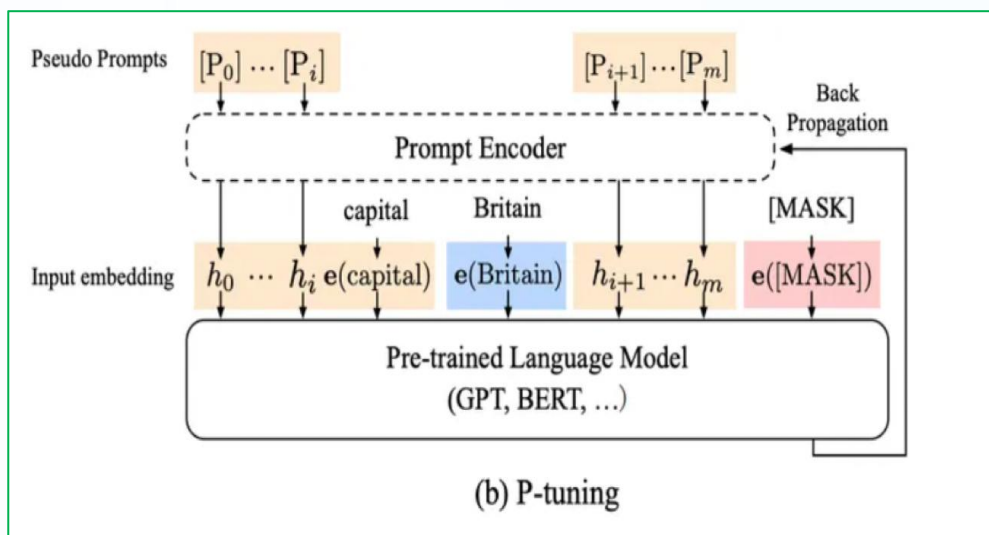
P-Tuning 提出将 Prompt 转换为可以学习的 Embedding 层，只是考虑到直接对 Embedding 参数进行优化。



## P-Tuning V1方法 (NLU任务)

P-Tuning V1 直接对 Embedding 参数进行优化会存在两个挑战:

- Discreteness(不连续性): 对输入正常语料的 Embedding 层已经经过预训练, 而如果直接对输入的 prompt embedding进行随机初始化训练, 容易陷入局部最优.
- Association(关联性分析): 没法捕捉到 prompt embedding 之间的相关关系.



用 MLP + LSTM 的方式来对 prompt embedding 进行一层处理.

$$h_i = \text{MLP} \left( \left[ \vec{h_i} : \overleftarrow{h_i} \right] \right) \\ = \text{MLP} ([\text{LSTM}(h_{0:i}) : \text{LSTM}(h_{i:m})])$$

P-tuning 固定 LLM 参数, 利用多层感知机 (MLP)和 LSTM 对 Prompt 进行编码, 编码之后与其他向量进行拼接之后正常输入 LLM. 注意, 训练之后只保留 Prompt 编码之后的向量即可, 无需保留编码器.



## I P-Tuning的特点

### 与Prompt-Tuning的区别

1

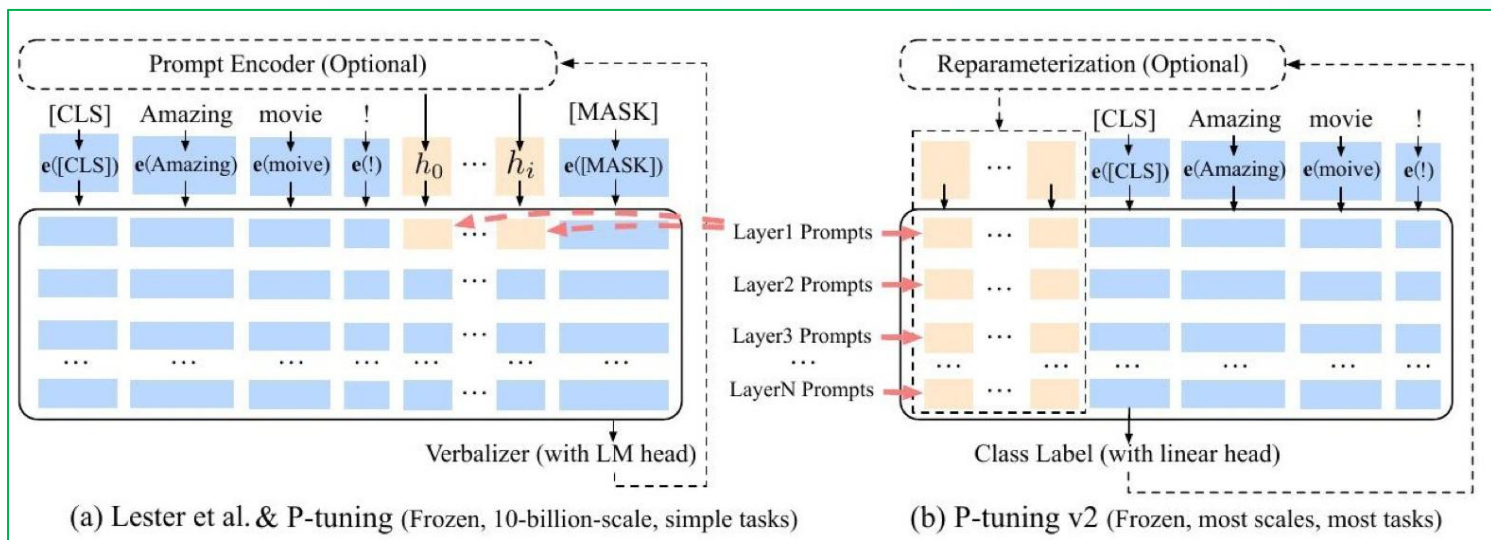
Prompt Tuning 是将额外的 embedding 加在开头，看起来更像是模仿 Instruction 指令；而 P-Tuning 的位置则不固定。

2

Prompt Tuning 不需要加入 MLP 来参数初始化；而 P-Tuning 通过 LSTM+MLP 来初始化。

## P-Tuning V2

P-Tuning V2是升级版本，主要解决P-Tuning V1 在小参数量模型上表现差的问题. 详细信息可参考 《[P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks]



P-Tuning v2 方法的核心思想：在模型的每一层都应用连续的 prompts, 并对 prompts 参数进行更新优化. 同时, 该方法也是针对 NLU 任务优化和适配的.



# 思考总结

Thinking summary

## 1.什么是Prompt-Tuning?

答案：通过添加模板的方法来避免引入额外的参数，从而让模型可以在小样本（few-shot）或者零样本（zero-shot）场景下达到理想的效果.

## 2.PET模型的主要组件?

答案：Pattern(模板)与Verbalizer(标签词映射).

## 3.P-tuning V1 的核心思想?

答案：P-tuning 固定 LLM 参数, 利用多层感知机 (MLP)和 LSTM 对 Prompt 进行编码，编码之后与其他向量进行拼接之后正常输入 LLM. 注意，训练之后只保留 Prompt 编码之后的向量即可，无需保留编码器



黑马程序员线上品牌



扫码关注博学谷微信公众号

