

2021-2022学年秋季学期

自然语言处理

Natural Language Processing



授课教师：胡玥

助 教： 李运鹏

自然语言处理
Natural Language Processing

第 7 章 预训练语言模型

2/2

授课教师：胡玥

授课时间：2021.11

内 容 提 要

7.1 预训练，精调范式 (Pre-train, Fine-tune)

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

■ NLP 范式发展:

P1. 非神经网络时代的完全监督学习--特征工程

(Fully Supervised Learning, Non-Neural Network)

特点: 人工进行大量的特征模版定义

P2. 基于神经网络的完全监督学习--架构工程

(Fully Supervised Learning, Neural Network)

特点: 人工设计各种网络结构

P3. 预训练, 精调范式 --目标工程 (Pre-train, Fine-tune)

特点: 引入各种辅助任务loss, 将其添加到预训练模型中, 然后继续pre-training, 以便让其适配下游任务, 之后, 通过引入额外的参数, 用特定任务的目标函数对模型进行微调, 使其更适配下游任务。研究重点转向了目标工程, 设计在预训练和微调阶段使用的训练目标

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

P4. 预训练, 提示, 预测范式--prompt挖掘工程 (Pre-train, Prompt, Predict)

Prompt is the technique of making better use of the Knowledge from the pre-trained model by adding additional texts to the input

特点: 不通过目标工程使预训练的语言模型 (LM) 适应下游任务, 而是将下游任务建模的方式重新定义 (Reformulate), 通过利用合适prompt实现不对预训练语言模型改动太多, 尽量在原始 LM上解决任务的问题

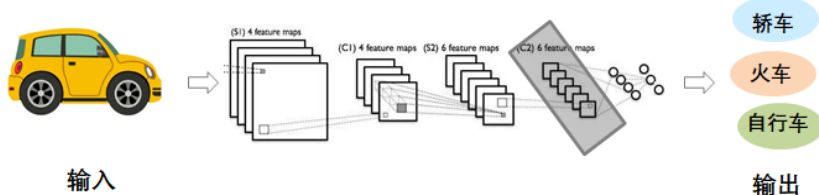
prompt learning激活了类似于小样本学习等场景

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

■ NLP 四个范式:

第一范式: 需要大量任务训练数据

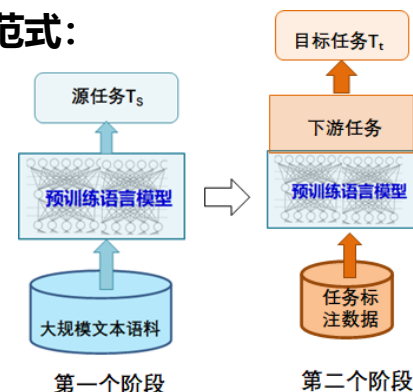
概率统计时代: 特征工程+算法 (Naive Bayes/SVM/LR/KNN……)



深度学习时代: 自动获取特征 (表示学习) 端到端分类

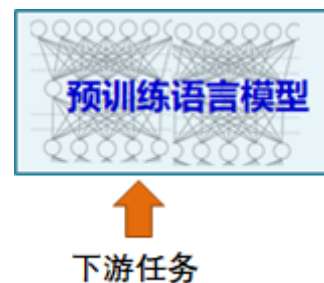
第二范式: 需要大量任务训练数据

第三范式:



大量预训练数据 + 少量任务数据

第四范式:



少量(无)的任务数据

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

■ 文本分类任务不同的建模范式

P2: 基于神经网络的完全监督学习)--架构工程
(Fully Supervised Learning, Neural Network)



关键问题: 如何生成高质量的文本表示

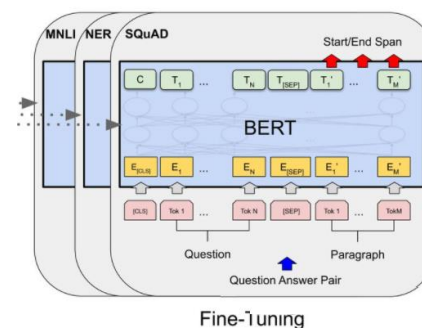
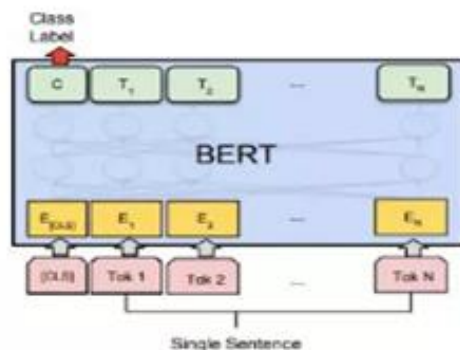
数据: 需要大量标注数据 (有监督)

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

P3: 预训练, 精调范式 (Pre-train, Fine-tune)--目标工程

通过预训练模型形成句表示, 然后将该句表示作为分类的输入

句表示



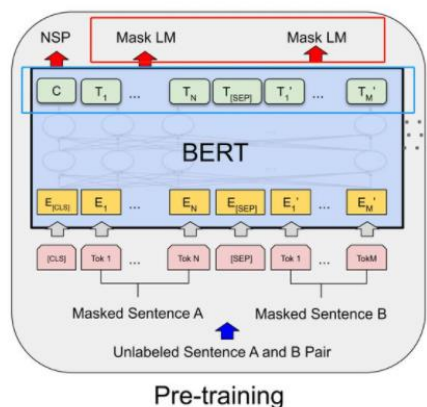
输入: 句子 $\{w_1, w_2, \dots, w_n\}$

数据: 需要领域数据进行任务微调

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

P4. 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict) --prompt挖掘工程

核心思想：改变任务形式利用预训练模型完成任务（用于小样本学习或半监督学习，某些场景下甚至能做到零样本学习。）

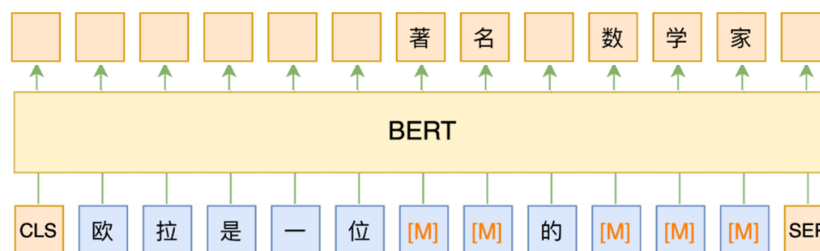
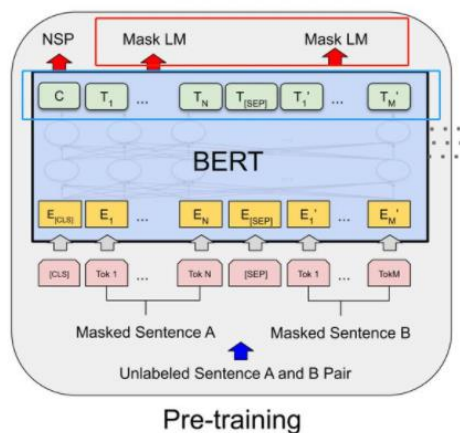


数据：少量（无）的任务数据

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

预训练模型 BERT 分析

BERT 采用 MLM 方式训练, 这种训练实际是填空形式训练, 所以BERT 见过大量的填空式知识, 如果能将任务转换为填空形式, 理论上可直接用 BERT 完成任务



怎么做?

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

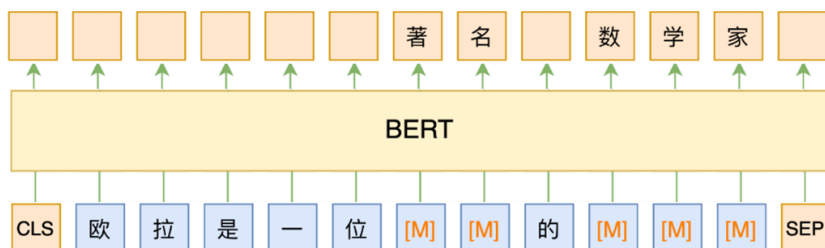
■ 分类任务 (二分类)

目标：对句子“这个餐厅的服务真不错。”情感倾向分类

任务输入：“这个餐厅的服务真不错。”

任务输出：任务标签 { 正向，负向 }

如何用BERT直接完成分类任务？



思路：将分类任务转换成完形填空问题

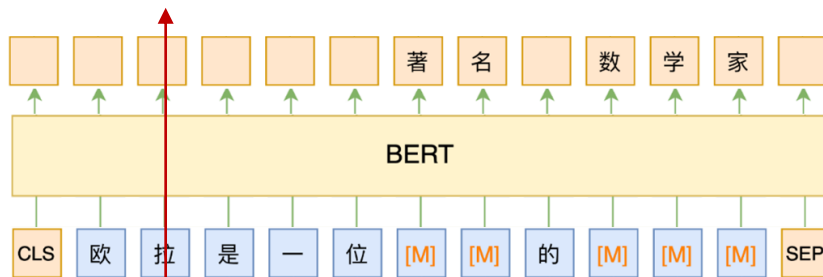
7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

方法：给输入的文本增加一个前缀或者后缀描述，并且Mask掉某些Token，转换为完形填空问题，转换要尽可能与原来的句子组成一句自然的话

如：给句子“这个餐厅的服务真不错。”补充描述（**补充模板**），构建如下的完形填空形式任务：

Prompt-任务输入： _____**满意。这个餐厅的服务真不错。**

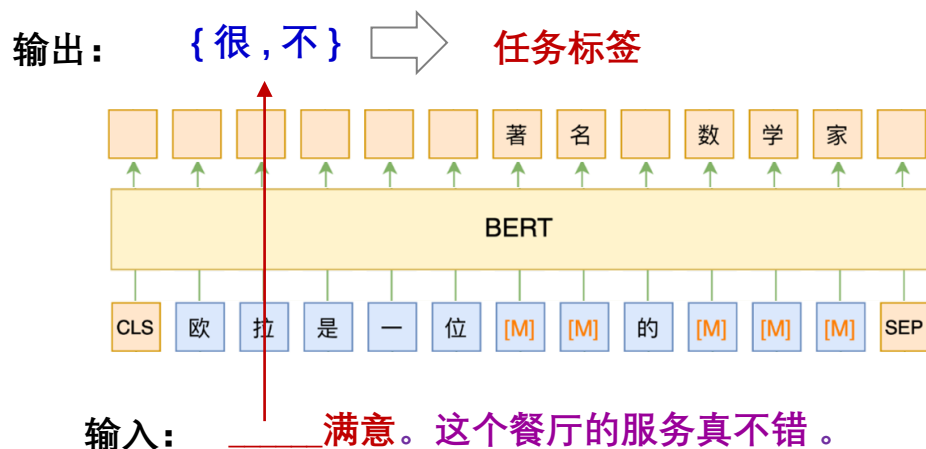
输出： {很, 不}



输入： _____**满意。这个餐厅的服务真不错。**

构建预测Token的候选空间,限制空位处只能填一个“很”或“不”，即要根据上下文一致性判断是否满意如果“很”的概率大于“不”的概率，是正面情感倾向，否则就是负面情感

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)



构建预测Token的候选空间后建立Token到实际类别的映射，定义一个verblizer作为映射函数，把label【正向】映射为“很”，把label【负向】映射为“不”。

BERT预训练时MLM任务是预测整个词表，而这里把词表限定在{很, 不}，交叉熵损失训练模型。预测时，假如预测出“很”样例预测label就为正向，预测“不”样例就是负向。

通过任务转换可以用MLM模型给出预测结果，而MLM模型的训练可以不需要监督数据，因此理论上这能够实现零样本学习。（如用少量任务样本微调一下模型，效果更好）

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

■ 分类任务 (多分类)

目标：对句子“滑雪运动员取得了很好的成绩。”主题分类

任务输入：“滑雪运动员取得了很好的成绩。”

任务输出：任务标签 { 体育，娱乐，访谈，影视 }

用BERT直接完成多分类任务

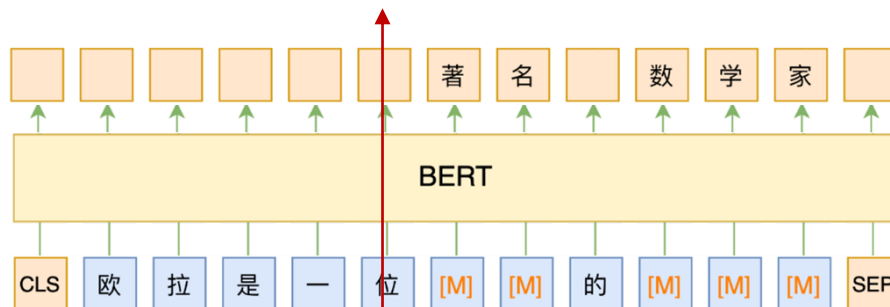
思路：将分类任务转换成完形填空问题

给句子“滑雪运动员取得了很好的成绩。”补充描述（补充模板），
构建如下的完形填空形式任务：

下面报导一则_____新闻。滑雪运动员取得了很好的成绩。

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

输出： { 体育，娱乐，访谈，影视 }



限制空位处只能填有
限个主题词，主题词
做为任务标签

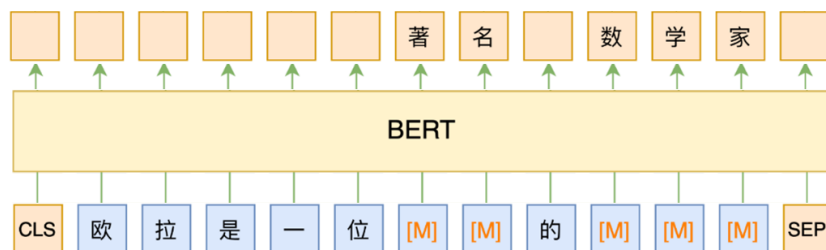
输入： 下面报导一则_____新闻。滑雪运动员取得了很好的成绩。

一个好的MLM模型应当能预测出“体育”二字来。

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

如: 匹配任务(蕴含)

目标: 判断这两个句子“**我去了北京**”跟“**我去了上海**”是否相容



方法: 将两个句子拼接起来输入到模型做, 作为一个二分类任务

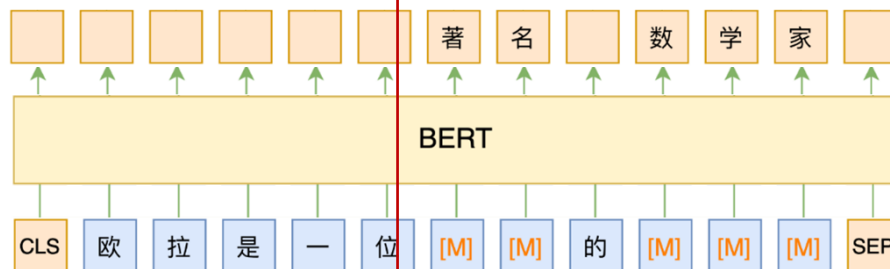
- 补充描述 (**补充模板**), 构建如下的完形填空形式任务:
我去了北京? _____, **我去了上海**。
- 其中空位之处的候选词为 {是的, 不是}
- 定义 tokenizer 映射函数, 把label **【相容】** 映射为“是的”, 把label **【不相容】** 映射为“不是”。

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

输出:

{ 是的, 不是 }

任务标签



输入: 我去了北京? _____, 我去了上海。

一个好的MLM模型应当能预测出二句子的关系。

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

基于MLM模型的局限性： MLM所使用的独立假设限制了它对更长文本的预测能力（空位处的文字不能太长），无法预测不定长的答案也约束了它的场景（所以当前只能用于做选择题，不能做生成）

生成任务可以用自回归的生成预训练模型做，原理相同

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

例如:

Few-shot
learning

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```

One-shot
learning

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```

Zero-shot
learning

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

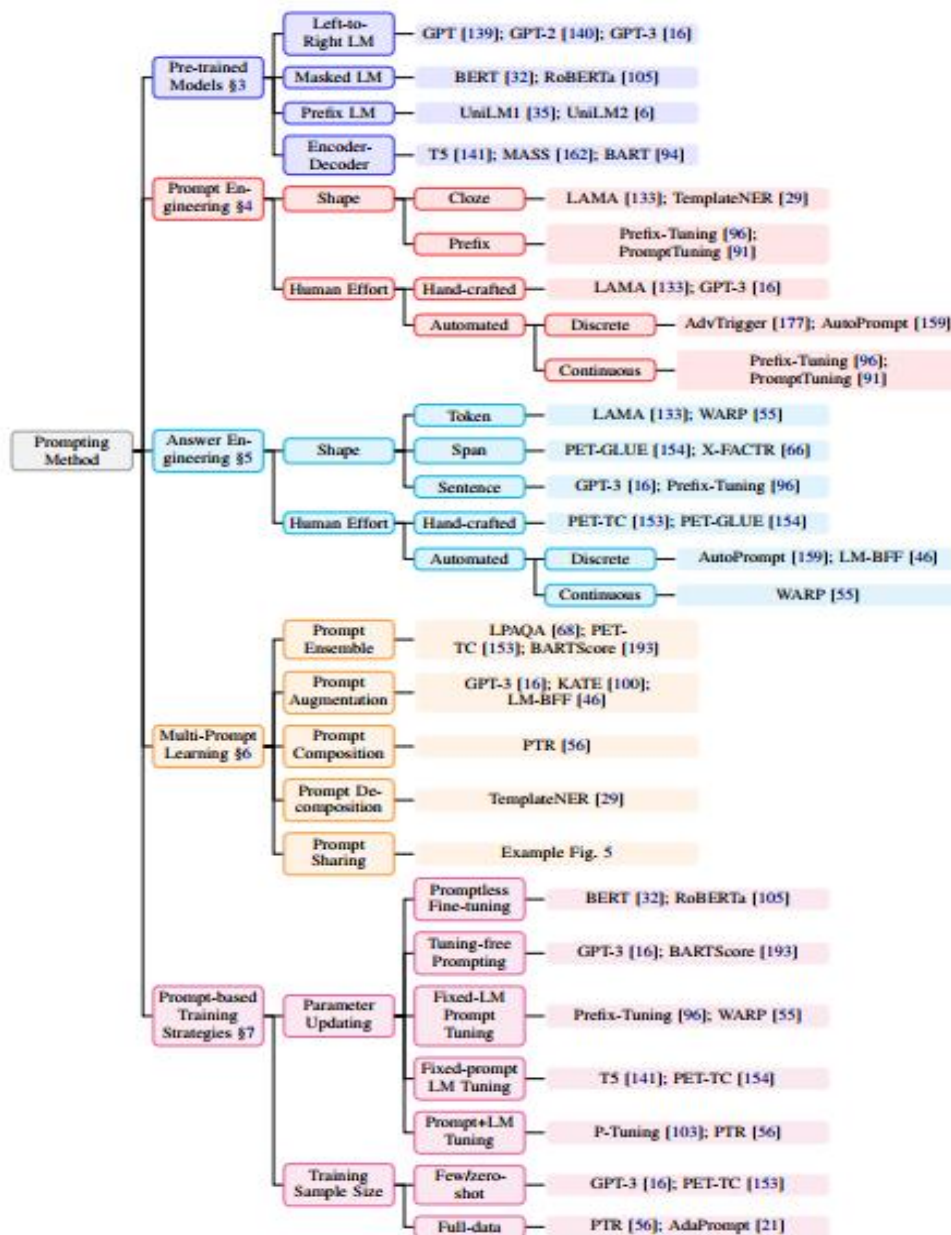
■ Prompt 主要问题:

1. **输入端**: 怎么样选取合适的Prompt, 适配不同任务, 同时把模型潜能激发出来 (怎么 把Prompt结合输入的句子构成模型的输入)
2. **输出端**: 模型的输出 (Answer) 可能与标签不同, 拿到answer后如何往标准化的Y (标签空间) 映射
3. **训练**: 怎样利用Prompt机制精调模型



7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

■ Prompt 研究导图:



7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

Prompt 基础

Prompt 方法：将原输入附加一段补充描述语句，通过这段补充描述语句实现任务转换和对任务求解，这段描述语句与原始输入一起要形成一段语义合理的语句作为prompt的输入

对于输入的文本 x ，进行两步操作形成 prompt 输入：

- 使用一个模板，模板通常为一段自然语言，并且包含有两个空位置：用于填输入 x 的槽 $[x]$ 和用于生成答案文本 $[z]$ 的槽
- 把输入 x 填到 $[x]$ 的位置

Name	Notation	Example	Description
<i>Input</i>	x	I love this movie.	One or multiple texts
<i>Output</i>	y	++ (very positive)	Output label or text
<i>Prompting Function</i>	$f_{\text{prompt}}(x)$	[X] Overall, it was a [Z] movie.	A function that converts the input into a specific form by inserting the input x and adding a slot [Z] where answer z may be filled later.
<i>Prompt</i>	x'	I love this movie. Overall, it was a [Z] movie.	A text where [X] is instantiated by input x but answer slot [Z] is not.
<i>Filled Prompt</i>	$f_{\text{fill}}(x', z)$	I love this movie. Overall, it was a bad movie.	A prompt where slot [Z] is filled with any answer.
<i>Answered Prompt</i>	$f_{\text{fill}}(x', z^*)$	I love this movie. Overall, it was a good movie.	A prompt where slot [Z] is filled with a true answer.
<i>Answer</i>	z	“good”, “fantastic”, “boring”	A token, phrase, or sentence that fills [Z]

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

Prompt Engineering(提示工程)

首先要考虑 prompt 的形式, 然后决定是采用手动还是自动的方式来创建所需形式的 prompt :

prompt的形式主要有二种:

1. 完形填空 (cloze) prompts: 用在如BERT-MLM式预训练模型上, 如情感分类任务可以输入「这个饼不错」, 太 [Z], Z 输出「棒」。一般 [Z]在句中
2. 前缀提示 (prefix) prompts: 用在如GPT2-3单向LM预训练模型上, 输入「好好学习」, 翻译成英文: [Z], [Z] 输出「good good study」.一般 [Z]在句末

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

选择哪种prompt 取决于任务和用于解决任务的模型。

- 对于有关生成的任务或使用标准自回归 LM 解决的任务，前缀 prompt 往往更有帮助，因为它们与模型从左到右的性质刚好吻合
- 对于使用掩码 (Mask) LM 解决的任务则完形填空 prompt 合适，因为它们与预训练任务的形式非常匹配
- 全文本重建模型则可以与完形填空 prompt 或前缀 prompt 一起使用。对于一些涉及多个输入的任务，如文本匹配，prompt 模板必须包含至少两个输入的空间。

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

模板创建有手工方式和自动化方式

手工模板工程: 手工制作模板的策略很直观, 且可以在一定程度上准确地解决各种任务, 但方法也存在一些问题: 创建和对这些 prompt 进行实验需要大量的时间和经验, 特别是对于一些复杂的任务, 例如语义解析等即使是经验丰富的 prompt 设计者可能也无法手工发现最佳的prompt

自动化模板设计: 自动化搜索的 prompt 可以进一步被分为离散 prompt (其中 prompt 是一个实际的文本字符串) 和连续 prompt (其中 prompt 直接在底层 LM 的嵌入空间中进行描述)。

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

Answer Engineering(答案工程)

Answer 工程的目的是搜索一个 answer 空间 Z 和一个到原始输出 Y 的映射, 从而得到一个有效的预测模型。需考虑的两个维度: 确定 answer 形式和选择 answer 设计方法。

◆ answer 形式

answer 的形式决定了它的粒度, 一些常见的选择包括:

- Token: 预训练 LM 词汇表中的一个 token, 或者词汇子集;
- Span: 短的 multi-token span, 这些通常与 cloze prompt 一起使用;
- 句子或文档: 这些通常与前缀 prompt 一起使用。

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

◆ answer 空间设计方法

设计适当的 answer 空间 Z , 以及如果 answer 不用作最终输出, 如何设计到输出空间 Y 的映射

方法: 手动创建 answer 和自动搜索 answer 的工作 (适用于离散 answer 空间和连续 answer 空间)

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

multi-prompt lending (复合模板学习)

前面探讨的 prompt 工程方法主要集中于为输入构建单个 prompt。大量的研究表明, 多重 prompt 可以进一步提升 prompting 方法的效果。常见的方法:

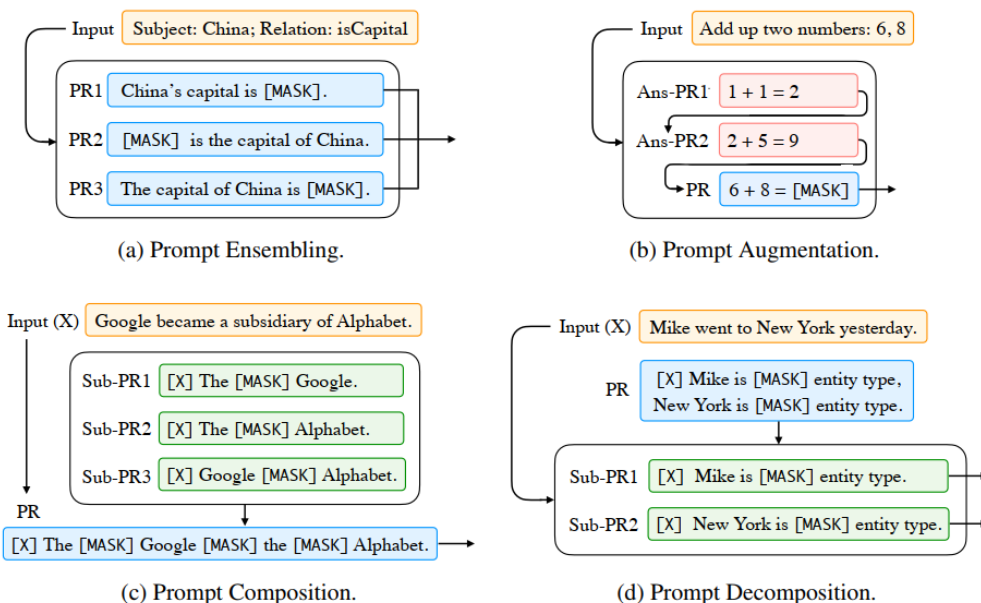


Figure 4: Different multi-prompt learning strategies. We use different colors to differentiate different components as follows. “” for input text, “” for prompt, “” for answered prompt. “” for sub-prompt. We use the following abbreviations. “PR” for prompt, “Ans-PR” for answered prompt, “Sub-PR” for sub-prompt.

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

(a) prompt ensembling

把多个prompt通过某种加权方法组合到一起。如图 (a)

(b) prompt augmentation

启发式学习。如图 (b)

(c) prompt composition

将复合的prompt句子, 拆解成多个小段prompt, 最后再组合在一起训练。如图 (C)

(d) prompt decomposition

由于一些任务的mask工作使用句子数量有限 (如词性标注任务) 固只能通过decomposition将一个句子拆分成多个部分后, 再对每个部分做prompt单独训练。如图 (D)

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

Prompting 方法的训练策略

训练设置

在很多情况下, 无需对下游任务的语言模型进行任何显式训练即可使用 prompting 方法, 这在传统上被称为零样本设置, 因为感兴趣任务的训练数据为零。

也有一些方法使用训练数据来训练与 prompting 方法一致的模型, 它们包括全数据学习或少样本学习。

prompting 方法在少样本学习中非常有用, 因为通常没有足够的训练示例来完全指定期望的行为, 因此使用 prompt 将模型推向正确的方向特别有效。

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

参数更新方法

在基于 prompt 的下游任务学习中，通常存在两种类型的参数，即来自预训练模型和 prompt 的参数。哪类参数应该更新是一项重要的设计决策，可以在不同场景中产生不同程度的适用性。

根据底层语言模型的参数是否需要调整、是否有额外的 prompt 参数和这些额外的 prompt 参数是否需要调整有 5 种调整策略：

- Promptless Fine-tuning
- Tuning-free Prompting
- Fixed-LM Prompt Tuning
- Fixed-prompt LM Tuning
- Prompt+LM Fine-tuning

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

prompt 参数调整策略：

Strategy	LM Params	Prompt Params		Example
		Additional	Tuned	
Promptless Fine-tuning	Tuned	-		ELMo [130], BERT [32], BART [94]
Tuning-free Prompting	Frozen	✗	✗	GPT-3 [16], AutoPrompt [159], LAMA [133]
Fixed-LM Prompt Tuning	Frozen	✓	Tuned	Prefix-Tuning [96], Prompt-Tuning [91]
Fixed-prompt LM Tuning	Tuned	✗	✗	PET-TC [153], PET-Gen [152], LM-BFF [46]
Prompt+LM Fine-tuning	Tuned	✓	Tuned	PADA [8], P-Tuning [103], PTR [56]

Characteristics of different tuning strategies. “Additional” represents if there are additional parameters beyond LM parameters while “Tuned” denotes if parameters are updated.

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

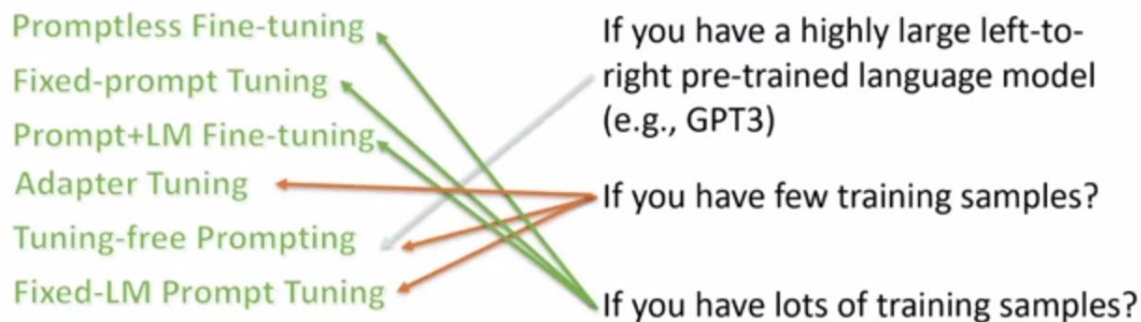
各策略特点:

- **Promptless Fine-tuning:** 直接使用下游任务进行训练, 不用prompt, 更新预训练参数; 优点-简单, 不需要设计prompt; 缺点-小数据集上容易过拟合或者不稳定
- **Tuning-free Prompting:** 基于prompt直接生成答案, 无需调参。优点-简单高效; 缺点-需要较重的prompt设计
- **Fixed-LM Prompt Tuning:** 固定预训练参数, 调整prompt参数, 使下游任务更好作用于prompt的产生; 优点-适合小样本学习; 缺点-不适合零样本学习, prompt模版通常不再是人工设计的
- **Fixed-Prompt LM Tuning:** 固定prompt参数, 调整预训练参数
- **Prompt+LM Fine-tuning:** 调整prompt参数, 调整预训练参数。适合大数据集, 小数据集易过拟合

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

策略选择:

Too many, difficult to select?



7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

Prompting 方法应用

根据研究者的调研, prompt 方法在以下诸多领域具有广泛的应用:

- 知识探索 (事实探索和语言学探索)
- 分类任务 (文本分类和自然语言推理)
- 信息提取 (关系提取、语义分析和命名实体识别)
- NLP 中的推理 (常识推理和数学推理)
- 问答
- 文本生成
- 文本生成的自动评估
- 多模态学习
- 元应用 (域自适应、除偏和数据集创建)

7.2 预训练，提示，预测范式 (Pre-train, Prompt, Predict)

Pattern Exploiting Training (PET)

核心思想：

将分类任务重构成填空形式；利用预训练模型和少量样本对无标签数据进行软标注；用软标注的数据训练分类模型，达到良好的效果

PET主要包括三个步骤：

- 对于每个pattern，分别使用一个语言模型（PLM）在小规模的数据集上进行微调；
- 集成所有的语言模型并为unlabeled数据集上进行标注，标注结果为soft-label（即每个label的概率分布）；
- 使用带有soft-label的数据训练分类器，使用分类器进行分类。

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

1. 参数说明:

假设 M 表示一个预训练语言模型, 词汇表记做 V , 其中包含一个 [MASK], L 表示目标任务 A 的所有标签集合。任务 A 的输入记做 $X = (s_1, s_2, \dots, s_k)$ 。其中 s_i 表示一个句子。如果 $k = 2$, 则输入的 X 是两个句子 (在自然语言推理或文本蕴含任务中使用)

定义函数 P : 其将 X 作为输入, 输出 $P(X)$ 表示将 X 转化为带有 [MASK] 的 phrase

定义映射 v : $L \rightarrow V$ 其表示将每个标签映射到具体的词。

(P, v) 作为 pattern-verbalizer-pair (**PVP**)

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

例如1: 其判断两个句子是否矛盾。输入则为两个句子

$\mathbf{x} = (\text{Mia likes pie}, \text{Mia hates pie})$

pattern映射为带有[MASK]的模板:

$P(\mathbf{x}) = \text{Mia likes pie? } \text{----}, \text{Mia hates pie.}$ “Yes” or “No”.

任务其原始的标签是entailment和contradiction, 根据 verbalizer v 可以设计映射关系为 {entailment: yes、contradiction: no}, 因此模型可以对[MASK]的部分输出预测yes或no的概率。

(P,v) 作为 pattern-verbalizer-pair (PVP)

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

例如2: PET设计Yelp用户评价星级预测的Prompt (a是句子) :

$$P_1(a) = \text{It was } _____. a \quad P_2(a) = \text{Just } _____! \parallel a$$

$$P_3(a) = a. \text{ All in all, it was } _____.$$

$$P_4(a) = a \parallel \text{In summary, the restaurant is } _____.$$

答案定义, 定义label到word的映射v, 在Yelp评价预测中, 定义了五个映射:

$$v(1) = \text{terrible} \quad v(2) = \text{bad} \quad v(3) = \text{okay}$$

$$v(4) = \text{good} \quad v(5) = \text{great}$$

(P,v) 作为 pattern-verbalizer-pair (PVP)

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

2. PVP训练与推理:

建模问题: 以往模型只要对 $P(l|x)$ 建模就好了 (l 是label), 现在加入了Prompt P 以及标签映射, 问题更新为:

$$s_p(l | \mathbf{x}) = M(v(l) | P(\mathbf{x}))$$

其中, $M(w|Z)$ 表示给定带有一个[MASK]标记的序列 Z , 语言模型可以在该 [MASK]位置填入词 $w \in L$ 的非归一化得分, s 相当于某个prompt下生成对应 word的logits。再通过softmax, 就可以得到概率:

$$q_p(l | \mathbf{x}) = \frac{e^{s_p(l|\mathbf{x})}}{\sum_{l' \in \mathcal{L}} e^{s_p(l'|\mathbf{x})}}$$

可使用cross-entropy进行训练

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

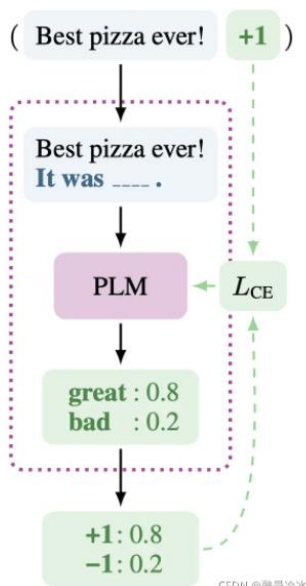
模型训练:

第一步: 先定义PVPs, 然后对每个PVP用labeled据进行单独的训练, 由于样本集合非常少, 会导致灾难性遗忘, 所以该步可以加入的Auxiliary Language Modeling一起训练

- 加入Auxiliary Language Modeling 训练方法:

样例1是labeled数据, 利用pattern把它改写后, 对__部分做完形填空预测 (即MLM任务)。样例2是一个unlabeled数据, 不对__部分做预测, 而是对被【MASK】做预测。这里的【MASK】可以采用BERT的方法, 随机对句子的15%token进行【MASK】。

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)



CSDN @ 图图冷冰

样例1: 这个餐厅的服务真不错。(label 为正向)

样例2: 我到这个商店买东西, 发现商品太差了(无label)

P(样例1): 这个餐厅的服务真不错。____满意

P(样例2): 我到这个[MASK] [MASK]买东西, 发现[MASK] [MASK]太差了____满意

$$L = (1 - \alpha) \cdot L_{CE} + \alpha \cdot L_{MLM}$$

训练时两个损失通过加权方式联合训练

好处: 能让model更适应于当前的任务

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

第二步：用训练好的PVPs，对unlabeled数据进行预测，并知识蒸馏，得到大量的soft label

如：有以下两个pattern，和两个verblizer，共4个PVP。如用他们对unlabeled数据进行预测（标注）由于无法知道用哪个PVP 效果好，所以采用对结果进行平均的方法标注获得 soft label

	V1（正向）=很	V1（负向）=不
P1:	a 。 <u>很</u> 满意	a 。 <u>不</u> 满意
P2:	<u>很</u> 好 ， a	<u>不</u> 好 ， a
	V2（正向）=太	V2（负向）=不
P1:	a 。 <u>太</u> 满意	a 。 <u>不</u> 满意
P2:	<u>太</u> 好 ， a	<u>不</u> 好 ， a

(P,v) 作为 pattern-verbalizer-pair (PVP)

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

用知识蒸馏方式解决：先用20个labeled数据训练4个PVP模型，然后拿这四个PVP模型对1K条unlabeled数据进行预测，预测的结果用下式进行平均。

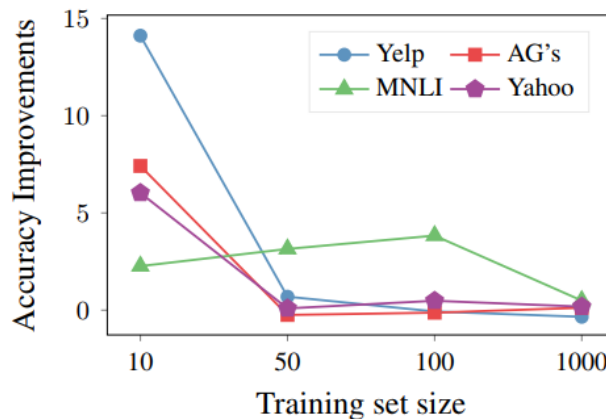
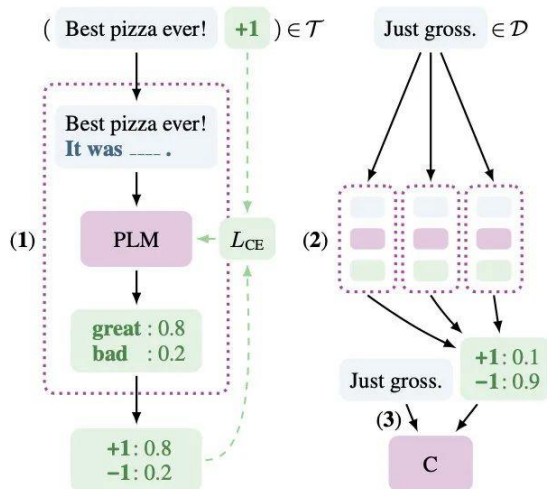
$$s_{\mathcal{M}}(l \mid \mathbf{x}) = \frac{1}{Z} \sum_{\mathbf{p} \in \mathcal{P}} w(\mathbf{p}) \cdot s_{\mathbf{p}}(l \mid \mathbf{x})$$

$s_p(l|x)$ 是单个PVP模型对样本预测的概率分布， $W(p)$ 是PVP的权重。最后对上式进行temperature=2的软化。利用训练好的PVPs所有1K条unlabeled数据打上soft label，

经过这样处理后利用多个PVP平均的思想把某些本来单个PVP预测偏差比较大的进行平均后修正，减少了噪声

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

第三步：再用第二步得到的1K条打上软标签的数据进行传统的有监督训练，训练完的model应用于下游任务的model



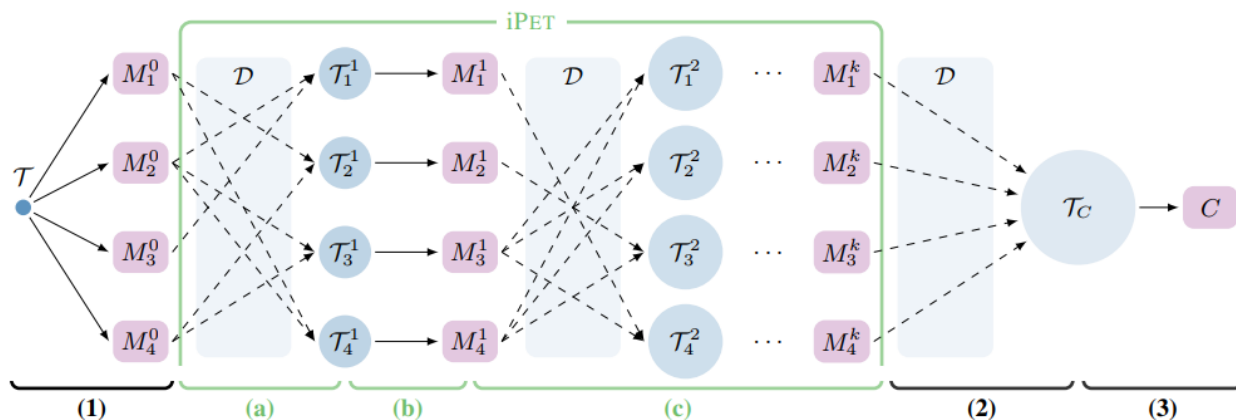
实验：训练集数量与效果提升的变化关系：

结论：PET在样本数量少的时候，基于Prompt的方法提升很明显，但是在数量较多时，相比传统方法只有小量提升。

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

多轮的 IPET

在每个PVP训练的过程中, 互相之间是没有互相交换信息没法相互学习, IPET通过迭代, 不断扩充上面训练PVP的数据集。逐步扩大soft label的范围, 这样新训的模型就能学到之前模型的知识



7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

模型效果

Line	Examples	Method	Yelp	AG's	Yahoo	MNLI (m/mm)
1	$ \mathcal{T} = 0$	unsupervised (avg)	33.8 \pm 9.6	69.5 \pm 7.2	44.0 \pm 9.1	39.1 \pm 4.3 / 39.8 \pm 5.1
2		unsupervised (max)	40.8 \pm 0.0	79.4 \pm 0.0	56.4 \pm 0.0	43.8 \pm 0.0 / 45.0 \pm 0.0
3		iPET	56.7 \pm 0.2	87.5 \pm 0.1	70.7 \pm 0.1	53.6 \pm 0.1 / 54.2 \pm 0.1
4	$ \mathcal{T} = 10$	supervised	21.1 \pm 1.6	25.0 \pm 0.1	10.1 \pm 0.1	34.2 \pm 2.1 / 34.1 \pm 2.0
5		PET	52.9 \pm 0.1	87.5 \pm 0.0	63.8 \pm 0.2	41.8 \pm 0.1 / 41.5 \pm 0.2
6		iPET	57.6 \pm 0.0	89.3 \pm 0.1	70.7 \pm 0.1	43.2 \pm 0.0 / 45.7 \pm 0.1
7	$ \mathcal{T} = 50$	supervised	44.8 \pm 2.7	82.1 \pm 2.5	52.5 \pm 3.1	45.6 \pm 1.8 / 47.6 \pm 2.4
8		PET	60.0 \pm 0.1	86.3 \pm 0.0	66.2 \pm 0.1	63.9 \pm 0.0 / 64.2 \pm 0.0
9		iPET	60.7 \pm 0.1	88.4 \pm 0.1	69.7 \pm 0.0	67.4 \pm 0.3 / 68.3 \pm 0.3
10	$ \mathcal{T} = 100$	supervised	53.0 \pm 3.1	86.0 \pm 0.7	62.9 \pm 0.9	47.9 \pm 2.8 / 51.2 \pm 2.6
11		PET	61.9 \pm 0.0	88.3 \pm 0.1	69.2 \pm 0.0	74.7 \pm 0.3 / 75.9 \pm 0.4
12		iPET	62.9 \pm 0.0	89.6 \pm 0.1	71.2 \pm 0.1	78.4 \pm 0.7 / 78.6 \pm 0.5
13	$ \mathcal{T} = 1000$	supervised	63.0 \pm 0.5	86.9 \pm 0.4	70.5 \pm 0.3	73.1 \pm 0.2 / 74.8 \pm 0.3
14		PET	64.8 \pm 0.1	86.9 \pm 0.2	72.7 \pm 0.0	85.3 \pm 0.2 / 85.5 \pm 0.4

经过一系列操作, 最终可以看到, PET在少样本的情况下完胜精调:

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

推荐使用的PVP

◆ 情感分析类型任务 (例如五分类的Yelp)

任务描述: 给定一个句子, 判断其情感类型, 例如电影评论等

pattern:

$P_1(a) =$ It was _____. a $P_2(a) =$ Just ____! || a

$P_3(a) =$ a . All in all, it was _____.

$P_4(a) =$ a || In summary, the restaurant is _____.

verbalizer:

$v(1) =$ terrible $v(2) =$ bad $v(3) =$ okay

$v(4) =$ good $v(5) =$ great

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

◆ 主题分类型任务 (例如AGNews、Yahoo)

任务描述: 给定一个句子, 判断其主题类型, 例如新闻分类等;

pattern:

$$P_1(\mathbf{x}) = \text{----}: a\ b \quad P_2(\mathbf{x}) = a\ (\text{----})\ b$$

$$P_3(\mathbf{x}) = \text{----} - a\ b \quad P_4(\mathbf{x}) = a\ b\ (\text{----})$$

$$P_5(\mathbf{x}) = \text{---- News: } a\ b$$

$$P_6(\mathbf{x}) = [\text{Category: ----}] a\ b$$

verbalizer:

可以直接将label class的词作为label word

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

◆ 句子对类型任务 (例如MNLI)

任务描述: 给定两个句子, 判断两个句子的相关性

pattern:

$$P_1(\mathbf{x}) = \text{"a"?} \parallel \text{----}, \text{"b"} \quad P_2(\mathbf{x}) = a? \parallel \text{----}, b$$

verbalizer:

$$v_1(0) = \text{Wrong} \quad v_1(1) = \text{Right} \quad v_1(2) = \text{Maybe}$$

$$v_2(0) = \text{No} \quad v_2(1) = \text{Yes} \quad v_2(2) = \text{Maybe}$$

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

■ 未来的挑战

- prompt的设计问题仍然是一个值得探讨的方向。目前使用Prompt的工作大多集中分类任务和生成任务, 其它任务则较少。如何针对不同的下游任务设计prompt?
- 模型的表现同时依赖于使用的模板和标签的转化, 如何同时搜索或者学习出两者联合的最好效果仍然很具挑战性。
- Prompt的理论分析和可解释性。尽管Prompt方法在很多情况下都取得了成功, 但是目前prompt-based learning的理论分析和保证还很少, 使得人们很难了解Prompt为什么能达到好的效果。
- 由于PLM在预训练过程中见过了大量的人类世界的自然语言, 所以很自然地受到了影响。比如太阳有几个眼睛。再比如说训练语料中有很多的"The capital of China is "Beijing.", 导致模型认为下次看到"capital" 的时候都会预测出"Beijing", 而不是着重看到底是哪个国家的首都。因此如何用prompt debias PLMs 学习到的偏置

7.2 预训练, 提示, 预测范式 (Pre-train, Prompt, Predict)

■ 相关工具

<https://github.com/thunlp/OpenPrompt>

清华发布了一个统一范式的prompt-learning工具包OpenPrompt（项目链接：<https://github.com/thunlp/OpenPrompt>），可以让初学者、开发者、研究者都能轻松地部署prompt-learning框架来利用预训练模型解决各种NLP问题。

参考文献:

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing

[Exploiting Cloze Questions for Few Shot Text Classification and Natural](#)

<https://zhuanlan.zhihu.com/p/395795968>

https://blog.csdn.net/qq_36426650/article/details/120788059

<https://zhuanlan.zhihu.com/p/375934846>

谢谢各位！



Q&A