

NLP方向总结-提示学习

结构：图1总结拓补图

论文

P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks

定义

传统监督学习

训练一个模型来接受输入x并预测输出y的P(y|x)  
LM适应于下游任务

基于 (prompting) 提示的学习

基于语言模型

将下游任务适应于LM

原始输入x被使用模板修改成一个文本字符串提示x', 其中有一些未填充的槽, 然后语言模型被用来概率性地填充未填充的信息, 得到一个最终的字符串x', 从中可以得出最终的输出y

例如, 当识别一个社交媒体帖子的情感时, "我今天错过了公交车。", 我们可以添加提示 "我感觉如此\_\_", 并要求LM用一个带有情感的词来填补空白。或者如果我们选择提示 "英语。I missed the bus today. 法语: \_\_", LM可能会用法语翻译来填空。

其它不同任务的表述形式见表3

SLM: 标准语言模型

自回归LM

例如GPT

CTR: 损坏的文本重建

对输入句子的噪声部分计算损失, 将处理后的文本恢复到未损坏的状态

例如BERT

FTR: 全文重构

通过计算整个输入文本的损失来重构文本

例如BART等翻译模型

L2R

从左到右的预训练方法

自回归式LM的一种, 它预测下一个单词

Mask

Mask的方式

目的是根据包围的上下文预测masked的文本片段

适用于cloze类型的提示任务  
适用于文本分类、自然语言推理, 和提取式问答

PLM

Prefix LM: 前缀语言模型

前缀LM是一种从左到右的LM, 它以前缀序列x为条件对y进行解码, 该序列由相同的模型参数编码, 但有一个全连接的mask

En-De

Encoder-decoder结构

预训练模型

表13进行了总结

缩写

提示工程

是创建一个提示函数fprompt(x)的过程, 该函数能在下游任务中产生最有效的表现

提示形式

填充提示

填补文本字符串的空白

前缀提示

延续字符串的前缀

有关生成的任务

多输入的任务

如文本对分类, 提示模板必须包含两个输入的空间, [X1]和[X2]

手动

提示模板

自动模板学习

离散方法(硬提示)-寻找合适的词汇

Prompt Mining

基于挖掘的方法, 在给定一组训练输入x和输出y的情况下, 自动寻找模板

提示语转述

采用现有的种子提示语(例如手工构建的或挖掘的), 并将其转述为一组其他的候选提示语, 然后选择在目标任务上达到最高训练精度的提示语

基于梯度的搜索

实际的token上寻找能够触发底层预训练的LM以产生所需的目标预测的短序列

提示语生成

将提示语的生成视为文本生成任务, 并使用标准的自然语言生成模型来执行这一任务

提示评分

使用单向LM对这些填充的提示进行打分, 选择具有最高LM概率的一个

连续方法(软提示)-寻找嵌入

前缀优化

将连续的特定任务向量序列预加到输入中的方法, 同时保持LM参数冻结

用离散提示语初始化的优化

用已经创建的提示语或用离散提示语搜索方法发现的提示语来初始化连续提示语的搜索

软硬提示混合优化

答案工程

寻找答案空间Z和与原始输出Y的映射

答案形式

Token

跨度

句子

答案空间设计

离散答案

答案解析

这些方法从初始答案空间Z'开始, 然后使用解析来扩展这个答案空间, 以扩大其覆盖范围

先修剪再搜索

首先产生一个由几个可信的答案Z组成的初始修剪过的答案空间, 然后一个算法在这个修剪过的空间上进一步搜索, 选择最终的答案集

标签分解

用于关系抽取

连续答案

多提示学习

多个提示提升性能(图4):  
(1) 利用不同提示的互补优势,  
(2) 减轻提示工程的成本, 因为选择一个表现最好的提示是具有挑战性的, (3) 稳定下游任务的性能。

基于提示的训练策略

参数更新

(i) 是否优化基础LM的参数,  
(ii) 是否有额外的提示相关参数, (iii) 如果有额外的提示相关参数, 这些参数是否被微调。

1. 微调需要在训练时存储所有参数的梯度和优化器状态, 需要更大显存

2. 微调的每个任务都需要有单独的模型参数

3. P-tuning v1超过100亿个参数时提示学习和微调性能才能相等

4. 提示学习序列标注任务中效果不好

1. 可控的参数量有限, 大多数语言模型目前只能支持512的最大序列长度(由于注意力的二次计算复杂性的成本), 所以填充的提示长度也是有限的。

微调VS提示学习

提示学习的问题

提示优化 (P-Tuning, Prompt tuning)

提示学习策略中, 冻结预训练模型参数, 去优化提示的方式。因为提示学习的好坏与提示词(或本文是提示嵌入)的好坏呈正相关。

离散提示: 即人工设计的提示

例如判断一个句子的情感: "很不错的电影", 设计提示: 它是[MASK], 输入变成: 它是[MASK], 很不错的电影,, 预测[MASK]的单词, 例如结果"好"或"坏", 即可知道情感是正向还是负向

嵌入提示: 即嵌入的提示

还是判断一个句子的情感: "很不错的电影", 提示变成了连续的嵌入向量: 那么整个输入也是一个向量, [提示的嵌入向量] + [MASK]+ [很不错的电影的嵌入向量]。那么我们优化提示的嵌入向量即可。

结构 (图2)

1. 通过前缀的方式添加

2. 在transformer结构的每层的前面添加连续的嵌入

对比只在嵌入层加提示, 这样训练优化更平稳  
添加接近输出层的提示比浅层提示效果更好

3. 对比第一代提示优化, 删除了带有LM头的verbalizers

使用CLS进行分类

1.连续提示的参数重新初始化对性能影响

4. 其它

2. 不同的提示长度对性能影响

任务越难, 提示越长更好

3. 多任务学习对性能影响