

自然语言处理中关于提示学习的研究进展

范森¹, 施水才^{1,2}, 王洪俊²

(1. 北京信息科技大学计算机学院, 北京 100101; 2. 拓尔思信息技术股份有限公司, 北京 100096)

摘要: 预训练语言模型的出现使处理自然语言处理任务模式发生了巨大变化, 对预先训练好的模型进行微调以适应下游任务成为目前自然语言处理任务的主流模式。随着预训练模型越来越大, 需要找到轻量化的替代全模型的微调方法, 基于提示学习的微调方法可以满足这一需求。对提示学习的研究进展进行总结, 先描述了预训练语言模型与提示学习的关系, 说明现在寻找替代传统微调方法的必要性, 然后详细解释了基于提示学习微调模型的步骤, 包括对提示模板的构建、答案搜索和答案映射, 再对提示学习在自然语言处理领域的应用举例, 最后对提示学习面临的挑战和未来可能的研究方向进行展望, 以期对自然语言处理、预训练语言模型和提示学习相关领域的研究提供参考借鉴。

关键词: 提示学习; 自然语言处理; 微调方法; 预训练语言模型; 深度学习

DOI: 10.11907/rjdk.231362

开放科学(资源服务)标识码(OSID):



中图分类号: TP391.1

文献标识码: A

文章编号: 1672-7800(2024)004-0215-06

Research Progress on Prompt Learning in Natural Language Processing

FAN Sen¹, SHI Shuicai^{1,2}, WANG Hongjun²

(1. School of Computer, Beijing Information Science and Technology University, Beijing 100101, China;

2. TRS Information Technology Co Ltd, Beijing 100096, China)

Abstract: The emergence of pre-trained language models has greatly changed the way natural language processing tasks are handled. Fine-tuning pre-trained models to adapt to downstream tasks has become the mainstream mode of natural language processing tasks. As pre-training models become larger and larger, it is necessary to find lightweight alternatives to full-model fine-tuning methods. Fine-tuning methods based on prompt learning can meet this demand. This article summarizes the research progress of prompt learning, first describing the relationship between pre-trained language models and prompt learning, explaining the necessity of finding alternatives to traditional fine-tuning methods, and then explaining in detail the steps of fine-tuning models based on prompt learning, including the construction of prompt templates, answer search and answer mapping. Then examples of the application of prompt learning in the field of natural language processing are given, and finally an outlook is given on the challenges and possible research directions faced by prompt learning, hoping this helps with research in natural language processing, pre-trained language models and prompt learning related fields.

Key Words: prompt learning; natural language processing; fine-tuning methods; pre-trained language models; deep learning

0 引言

近几十年来,随着互联网的快速发展,人们的生活方式发生了巨大变化,每天互联网上会产生数量巨大的文本信息。要从大量文本信息中快速获取所需信息,就需要借助自然语言处理(Natural Language Processing, NLP)技术。自然语言处理是计算机科学与语言学相交叉的领域,也是人工智能领域一个重要的研究方向。该领域的主要目标

是研究如何让计算机更好地理解人类语言,并能够自然地与人类进行交流。

预训练模型的出现大大推动了自然语言处理领域的研究和发展,预训练模型采用的是两阶段的学习方法,先在大语料库中进行训练,让模型学习通用的语言表示,再根据具体下游任务对模型进行微调。但是随着模型数据规模的增加,如果使用传统的微调方法需要消耗巨大的存储资源,对计算机算力也有很高要求,因此预训练模型的微调方法也成为目前自然语言处理领域的一个重要研究

收稿日期: 2023-04-06

作者简介: 范森(1996-),男,北京信息科技大学计算机学院硕士研究生,研究方向为自然语言处理;施水才(1966-),男,硕士,北京信息科技大学计算机学院教授、硕士生导师,研究方向为信息检索、大数据分析、人工智能。本文通讯作者: 施水才。

方向。其中,基于提示学习(prompt learning)的微调方法是现在主流的微调方法之一。基于提示的微调方法可以利用模板将下游任务转换为与预训练模型相匹配的形式,从而缩小预训练与微调训练之间的差距,进而提升下游任务的表现^[1]。因此,提示学习对于NLP领域具有重要意义。对于NLP领域的学者和开发者而言,了解和研究提示学习技术,不仅可以提高模型的性能和泛化能力,而且可以在更广泛的场景中实现更加有效的NLP技术应用。

本文通过回顾和总结自然语言处理领域内提示学习的最新研究成果,综合分析提示学习的关键技术、应用实例及其潜在问题,旨在梳理该研究领域的发展脉络,并展望未来可能的研究方向。

1 预训练语言模型与提示学习

一直以来,完全监督学习在自然语言处理任务中发挥着核心作用,即模型只在目标任务输入—输出示例的数据集上进行训练,但这种手动标注数据集对于训练高质量的模型是远远不够的^[2]。随着深度学习的发展,越来越多神经网络模型应用于自然语言处理领域,显著特征的学习与模型本身的训练一起进行,重点转移到架构工程,通过设计合适的网络架构来提供归纳偏差,更有利于学习这些特征^[6]。

自2018年起,迁移学习在自然语言处理中得到了广泛应用,导致NLP模型的学习发生了巨大变革,完全监督学习的作用变小,标准转到了预训练和微调范式上。在此范式中,具有固定体系结构的模型被预先训练为语言模型(Language Model, LM),用于预测文本序列中下一个词的出现概率。训练语言模型使用的原始文本中包含大量信息,因此可以利用大规模的数据集训练这些语言模型,学习语言的健壮通用特性,然后通过引入额外的参数,并使用任务特定的目标函数对模型进行微调,使预先训练好的语言模型适应不同的下游任务目标。目前已经出现了许多预训练语言模型,包括ELMO^[7]、BERT^[8]、GPT^[11]、XLNET^[9]、Albert^[10]、ELECTRA^[12]等模型。预训练语言模型大多采用Transformer^[13]结构,例如现在广泛使用的BERT模型就是基于多层双向Transformer编码器实现的。双向结构的主要优点在于可以利用前后文信息,即同时考虑当前词语的左侧和右侧语境,从而更好地捕捉文本的上下文语义。该方式可以避免仅考虑当前词语而导致的歧义和误判情况,从而进一步提高模型性能和任务完成的准确性。

随着计算机算力的提升,预训练模型参数也在快速增长,例如从GPT模型2018年的1.17亿参数到GPT-3模型2020年的1750亿参数,增长了近1500倍且还在增加。增加模型参数可以提高应用效果,但也带来一些问题。例如庞大的参数会导致模型迁移能力下降,对于千亿参数的大

型模型来说,传统的预训练微调方法很难适用。即使在微调过程中使用较大的数据集,也很难保证模型能否快速学习到微调样本^[14]。此外,使用传统的微调方式会对存储资源造成巨大压力,对算力也有更高要求,很难在实际场景中部署应用^[1]。

近几年,自然语言处理领域的研究学者们提出了许多传统微调方法的替代方法,基于提示学习的微调是主流方法之一。在该范式中,不再是使预训练好的LM适应下游任务,而是重新制定下游任务,使之更贴近语言模型原本训练阶段所面对的问题类型。例如,对于句子“我今天在路上绊倒了”,可以通过添加提示“我感觉到[MASK]”来构建输入,放入到LM中要求用一个情感词来填补[Mask]。通过该方法,选择适当的提示模板即可操作模型的行为,以便预先训练的语言模型本身可用于预测所需的输出,在某些情况下可避免针对特定任务进行额外的训练^[15]。使用基于prompt的微调的优点是在给出一套合适的prompt模板的情况下,可以完全无监督的方式训练一个LM完成多个任务^[16]。

2 基于提示学习的微调方法

提示学习方法的原理是通过为不同的下游任务设计模板,将下游任务转换成与模型训练时相似的形式,从而减少预训练与下游任务微调时的差距。采用基于提示的方法微调可以有效减少预训练模型在下游任务微调过程中的资源存储和运算需求。prompt方法有3个步骤:

(1)模板设计。首先需要设计一个模板,将输入的 x 转变成 x' 。一般情况下 x' 中包含空槽,让预训练模型对空槽进行填充。模板的设计有很多种,需要根据具体下游任务的类型和具体使用的预训练模型来选择合适的模板。如图1所示,输入 x :“我喜欢这本书”,选择一个模板:“这是一本[Mask]书”。其中,[Mask]是空槽,经过prompt函数得到 x' :“我喜欢这本书,这是一本[Mask]书”。

(2)答案搜索。经过步骤(1)得到新的输入 x' 后,将 x' 放入训练好的LM中,得到可能的答案词表,例如“精彩的”“好”“经典的”“糟糕的”等词,在答案空间中找到得分最高的答案并放到空槽中。

(3)答案映射。在空槽填入答案后,一些任务需要对空槽的内容进行转换(例如分类任务),将转换后的内容作为最终的输出标签 y ,如图1中最终输出积极的标签。而一些任务可以把填入空槽的答案作为输出,不需要进行答案映射。

2.1 模板设计

提示学习研究过程中最重要步骤之一是如何设计prompt模板进行微调,以提升下游任务的表现。目前,主要的模板设计方法有3种,具体如下:

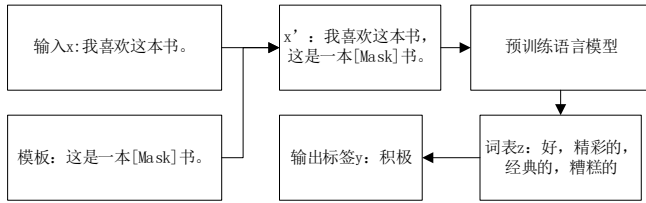


Fig. 1 Prompt flow
图1 Prompt 流程

2.1.1 人工构建模板

创建提示模板最直观、高效的方法是依靠专业人员的经验手动创建,使用人工构建模板即使使用大规模的预训练语言模型也可以实现小样本甚至零样本学习,同时可以一定程度上完成各种任务。例如, Petroni 等^[17]在 2019 年以 BERT 模型作为基础,探测预训练模型中可能包含的知识。其使用类似掩码语言模型 (Masked Language Model, MLM) 预训练机制的方法,将下游任务的训练数据集转换成完形填空形式 (见表 1),以缩小预训练时与微调时所涉及数据集的差异,从而可更好地检测模型中所包含的知识。此外,其还提出可用于评估预训练语言模型中所包含知识的 LAMA probe 数据集。LAMA probe 数据集整合了多个数据集 (包括 T-Rex、Google-RE 等),通过构建模板将知识三元组转换成完形填空形式,以探测其中包含的知识。该思想给后来的提示学习研究者们很大启发。

Table 1 Template construction example
表 1 模板创建举例

| 问题 | 答案 |
|---------------|-------|
| 中国的首都是__ | 北京 |
| 中国国庆节在__月 | 十 |
| 小说《红高粱》的作者是__ | 莫言 |
| 端午节的起源国是__ | 中国 |
| 地球最高峰是__ | 珠穆朗玛峰 |

Open AI 在 2020 年发布了 GPT-3,该模型也采用人工构建模板的方式来解决下游任务的问题^[18]。GPT-3 首先冻结预训练好的模型参数,然后在输入的文本中加入提示信息构成模板,从而引导模型完成相应问题,而无需为多个下游任务分别设计模型,使得同一模型可以服务于不同任务。通常,这些任务提示由任务描述和几个相关示例组成,将输入文本和任务提示拼接成一个新的输入,以避免进行全模型微调,并获得了良好的结果。然而,GPT-3 也存在一些缺点。首先,GPT-3 不能提取上次预测时的有用信息并储存,导致每次作新预测时都需要重新给定任务的训练样本。其次,该方法利用注意力机制处理序列信息并基于上下文进行学习,但由于模型输入序列通常具有固定长度,导致无法充分发挥大数据集的优势。最后,GPT-3 有 1 750 亿参数,模型参数量巨大,很难在实际场景中部署应用。

虽然人工构建模板的方法直观、高效,可以一定程度上完成各种任务,但也存在几个问题:①创建模板需要消

耗大量时间,也需要构建者具有丰富的经验,特别是一些比较复杂的任务;②即使是经验丰富的构建者可能也无法手动发现最佳的提示模板;③构建过程比较繁琐,而且即使是轻微的模板变化也可能导致结果出现显著变化。

2.1.2 离散模板构建

针对人工构建模板存在的问题,NLP 学者们提出了很多方法来自动化模板设计过程,并将其进一步分为离散模板构建和自动化模板构建。离散模板构建是一种自动搜索、用于描述离散空间中模板的方法,这些模板由特定字符组成。

2020 年,Shin 等^[20]提出 AUTOPROMPT 来解决人工构建模板性能不稳定,同时需要大量时间构建的问题,通过梯度引导搜索为不同任务创建提示。该搜索以迭代的方式完成,在提示符中逐级查找标记。AUTOPROMPT 先定义一个带有触发印记的 X_{trig} 、原始输入 X_{inp} 和空槽 [MASK] 的模板 λ 。其中,由梯度搜索来确定 X_{trig} ,在所有任务中共用,空槽 [MASK] 用于预测输出结果。将 X_{trig} 和 X_{inp} 输入到模板中,生成的 X_{prompt} 作为预训练模型的输入。其采用两阶段操作自动选择标签集合的方法,以解决抽象任务中标签选择不清晰的问题。先训练一个逻辑分类器,将 [MASK] 作为预测类标签的输入,然后将预训练模型的输出输入到训练好的逻辑分类器中,以获得相应标签。相较于人工构建模板,采用 AUTOPROMPT 可以有效减少人力和时间成本,同时获得了不错的性能。

采用 GPT-3 在小样本上进行微调取得了很好的效果,但由于参数太多,不适用于真实场景。受到 GPT-3 的启发,Gao 等^[21]提出一种小样本微调预训练模型 (LM-BFF)。GPT-3 采用的人工标注方法可能导致局部最优解问题,LM-BFF 使用 T5 模型自动创建提示模板^[22]。模板生成步骤如下:①首先给定一些候选标签词,例如好或差等,这些词可以代表二分类任务中的标签;②在每个标签词前后添加填充位,随后输入到 T5 模型进行处理,自动生成对应序列;③将标签词转换为 [MASK] 标签,形成多个模板;④使用集束搜索来解码生成多个候选模板;⑤最后利用 dev 集对每一个候选模板进行微调 and 评估,选择其中最优的一个作为最终的文本模板。模板生成步骤如图 2 所示。需要注意的是,尽管 LM-BFF 的微调效果优于标准微调方法,但仍然落后于基于全量标注数据的标准微调方法。此外,该方法只适用于少量类别,句子模板的长度受到限制,其结果中也存在着较大方差。

2.1.3 连续模板构建

离散模板构建相对于人工设计方法简单且可以自动生成,但在许多情况下性能不够理想。提示构建的目的是为了找到一种能够让语言模型有效执行任务的方法,因此并不需要提示限制在人类可以理解的自然语言之内。相较而言,连续模板构建方法直接在模型的嵌入空间执行提示,摆脱了自然语言词嵌入的限制,并且消除了预先训

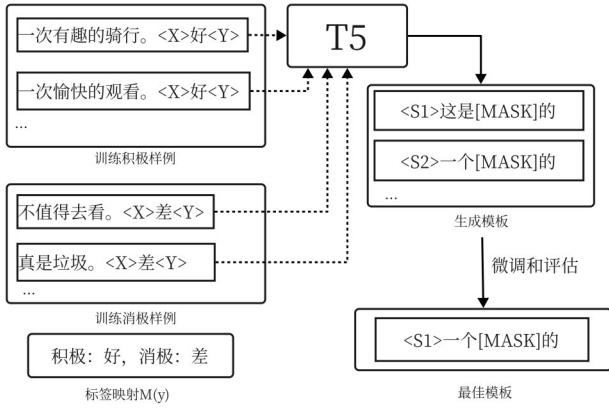


Fig. 2 Template flow of LM-BFF generating prompt

图2 LM-BFF生成提示模板流程

练的语言模型参数化设计的限制。此外,该方法可以强调语义,对具有误导性和歧义性的词语进行调整。

全模型微调时每个任务要存储一份预训练模型参数,针对这一问题, Li等^[24]在2021年提出 prefix-tuning 方法。Prefix-tuning 是一种基于预训练模型的生成模型,其核心思想是在预训练语言模型的每一层中插入与任务相关的向量,并通过调整下游任务的相关参数实现对生成结果的控制。同时,为了保证模型的稳定性和鲁棒性,Prefix-tuning 会冻结预训练模型参数。因此,即使在小规模的数据集上,Prefix-tuning 也能够获得比传统微调方法更好的效果。自回归语言模型在添加 prefix 之后的表示为 $z = [prefix; x; y]$, Encoder-Decoder 模型表示为 $z = [prefix; x; prefix; y]$ 。模型添加前缀可以训练的参数为 $P_{\theta} \in R^{|p_{idx}| \times \dim(h_i)}$, 其中 $|p_{idx}|$ 表示前缀长度, $\dim(h_i)$ 表示模型在第 i 时刻隐藏状态的维度。Prefix 的长度可以自定义,但其宽度必须与预训练模型隐藏状态的维度保持一致。对以下对数似然目标进行梯度更新,最终训练目标为:

$$\max_{\phi} \log p_{\phi}(y|x) = \sum_{i \in Y_{idx}} \log p_{\phi}(z_i | h_{<i>i}) \quad (1)$$

其中, p_{ϕ} 是可训练的语言模型分布, $h_{<i>i} = [h_{<i>i}^{(1)}; \dots; h_{<i>i}^{(n)}]$ 是时间步长 i 处所有神经网络层的串联。时间步长如果在 前缀内,则直接从前缀矩阵 M_{ϕ} 中复制,否则使用预训练的语言模型进行计算。Yidx 表示与 y 对应的指标序列。实验中发现这种基于前缀的连续模板比使用实词的离散模板对不同的初始化更敏感。

Lester 等^[25]在 prefix-tuning 的基础上进行了简化,并提出 Soft-Prompt 方法。其使用特殊标记预先设置输入序列,以形成模板并直接调整这些标记的嵌入。Soft-Prompt 与 prefix-tuning 相比,增加的参数更少,因为其不会在每个网络层中引入额外的可调参数。

Hambardzumyan 等^[26]受到对抗性重新编程^[27]的启发,提出了 WARP (Word-level Adversarial Reprogramming), 一种基于对抗性重构的替代方法,扩展了早期的自动提示生成工作。对抗性重构试图学习任务特定的词嵌入,指示语

言模型解决指定任务。与人工构建和离散模板构建不同, WARP 直接优化词嵌入作为提示,给模型提供更大的灵活性,并最终在下游任务中实现更出色的表现。在该模型中需要调整的参数只有两组 embedding $\Theta = \{\Theta^P, \Theta^V\}$, 其中 P 代表 Prompt, V 是对每一类的分类参数,类似于全连接层。将 Prompt token $P_{1 \dots p}$ 插入到输入序列中的任意位置,经过编码器和掩码语言模型头之后,最终通过 Θ^V 进行处理。分类概率通过以下公式进行计算:

$$P_{\Theta}(y|x) = \frac{\exp \Theta_y^V f(T_{\Theta^P}(x))}{\sum_{i \in C} \exp \Theta_i^V f(T_{\Theta^P}(x))} \quad (2)$$

其中, T_{Θ^P} 是插入了 prompt Θ^P 的序列, C 是所有类别, $f(x)$ 是预训练语言模型的输出,最后利用梯度优化算法找到能使 loss 最小的参数。

$$\Theta^* = \arg \max_{\Theta} (-\log P_{\Theta}(y|x)) \quad (3)$$

Table 2 Comparison of template construction methods

表2 模板构建方式比较

| 模板设计方法 | 优点 | 缺点 |
|--------|--|------------------------------------|
| 人工构建模板 | 依靠专业的经验手动创建模板更直观、高效,可以有效提高预训练模型在下游任务中的表现 | 构建过程繁琐,需要构建者具有丰富的经验,难以发现最佳模板,性能不稳定 |
| 离散模板构建 | 模板构建简单,可以自动生成,使用者不需要太多专业知识 | 在部分任务中效果不够理想 |
| 连续模板构建 | 直接在模型的嵌入空间中执行提示,不受自然语言的约束,可以根据来自下游任务的训练数据进行微调,效果理想 | 训练模型需要较长时间,并且在更换预训练模型后需要重新进行训练 |

2.2 答案搜索与映射

创建完模板得到 x' 后,接下来搜索使 LM 得分最大化的文本 \hat{z} 。首先定义 z 的允许值 Z , Z 在生成任务时可以是所有语言,在分类任务时可以是语言中的一部分。例如定义 Z 为 {"优秀"、"好"、"一般"、"差"、"糟糕"} 来表示 $Y = \{++, +, \sim, -, --\}$ 中的每个类,然后定义一个函数 $f_{fill}(x', z)$, 用可能的答案 z 来填充 x' 中的 [MASK] 位置。将通过该过程的所有提示作为填充提示 (filled prompt)。特别的,如果提示中填写的是真实答案,将称其为已回答提示 (answered prompt)。最后,使用预先训练的 $LMP(\cdot; \theta)$ 计算相应填充提示的概率来搜索潜在答案集 z 。

$$\hat{z} = \underset{z \in Z}{\text{search}} P(\text{fill}(x', z); \theta) \quad (4)$$

该搜索函数可以是搜索得分最高的 argmax 搜索,也可以是根据 LM 的概率分布随机生成输出采样。

最后,需要从得分最高的答案 \hat{z} 得到得分最高的输出 \hat{y} 。在某些情况下,答案本身就是输出,如文本生成和翻译任务等。在有些情况下,多个答案也会导致相同的输出,例如情感分类采用多个情感词来表示同一类别。在此情况下,有必要在搜索的答案与输出值之间进行映射。

3 提示学习在自然语言处理中的应用

提示学习给定一组合适的 prompt,以完全无监督的方式训练的单个语言模型即能用于解决大量任务。其在自然语言处理中有着广泛应用,包括分类任务、信息提取、文本生成和问答任务等方面。

3.1 分类任务

在分类任务中,提示模板的构建相对容易,例如文本分类和文本对分类。基于提示的分类任务关键是将其重新定义成一个合适的提示。Yin等^[28]使用一个提示符,例如“这篇文档的主题是[MASK]”,然后放入到预训练的掩码语言模型中进行填充。在文本分类中,以前的大部分工作都使用完形填空提示,在使用提示学习微调后,证明了提示学习对文本分类的有效性。文本对分类任务的目的是预测两个句子之间的关系,包括相似度和包含关系等。典型的文本对分类任务包括文本匹配、自然语言推理和文本相似预测等。文本对分类与文本分类相似,大多使用完形填空提示。对于模板设计,研究者主要关注的是少样本背景下的模板搜索,答案空间 Z 一般是从词汇里手动预先选择的。

3.2 信息提取

在分类任务中,完形填空问题通常可以直观构造出来。但是与分类任务不同,在信息提取任务中,构造提示往往需要一些技巧。

关系抽取是一项预测句子中两个实体之间关系的任务,2022年。Chen等^[29]首次探究了基于提示的LM微调在关系抽取中的应用,总结出从分类任务中继承提示方法的两个挑战:第一个挑战是关系抽取需要更大的标签空间,导致答案搜索更加困难,例如情感分类的标签空间需要两个,而关系抽取任务可能需要几十个标签;第二个挑战是在关系抽取任务中,每个标签的重要程度不同,但在原始的提示模板中,每个词的重要性被视为相同,对分类任务影响不大。对于这两个问题,Chen等提出一种自适应的答案选择方法来应对第一个挑战。而对于第二个挑战,其构建面向任务的提示模板,使用特殊标记来突出模板中的实体。

命名实体识别(Named Entity Recognition, NER)是指在给定的句子中识别人名、地名等实体的任务。与分类不同,基于提示的NER的困难在于要预测的每个单元是一个 token 或者片段,不是整个输入文本,并且样本上下文中的 token 标签之间也可能存在潜在关系。提示学习在标记任务中还没有得到充分探索,Cui等^[30]提出一种基于模板使用BART的NER模型,该模型枚举了文本跨度,并考虑了人工构建模板中每一种类型生成的概率。例如,给定一个输入“小明今天去北京”,使用“小明是一个[MASK]实体”来确定实体“小明”的类型,答案空间 Z 由“人名”“地名”等

值组成。

3.3 文本生成

文本生成是指根据给定条件生成符合条件文本的任务,通常以给出的信息作为条件。通过 prefix-tuning 和自回归预训练语言模型可以将基于提示的方法轻松地应用于这些生成任务中。Radford等^[31]首次展示了提示学习在文本生成任务中的强大潜力。其通过实验证明了在处理如机器翻译和文本摘要生成等复杂的文本生成任务时,利用提示学习法能够提升模型的性能和文本生成质量。

3.4 问答任务

问答任务是指对给定的输入问题作出回答。问答任务有多种,包括从包含答案的上下文文档中识别内容的抽取式问答、从多个选项中选择答案的多选式问答和返回任意字符串作为响应的自由形式问答。通常需要使用不同框架处理这些不同格式的任务,但是使用预训练语言模型和提示学习的好处是在同一框架下处理不同格式的问答任务。例如2020年,Khashabi等^[32]通过微调基于 seq2seq 的预训练语言模型,并利用来自上下文和问题的恰当提示,使许多问答任务变成了文本生成问题。

4 未来发展趋势与面临的挑战

提示学习在不同的任务和场景中都显示出巨大潜力,相信未来可以应用于更多领域,但是提示学习同样面临一些挑战。首先是预训练语言模型的选择,目前有大量预先训练好的语言模型可供选择,如何选择预训练语言模型,以更好地利用提示学习完成自然语言处理任务也是一个值得思考的问题。其次,目前关于提示学习的工作大多围绕着分类任务和文本生成任务展开。由于模板设计比较困难,信息提取和文本分析任务现在很少被讨论,未来在将提示学习应用于这些领域时,可能需要对这些任务重新进行定义,然后借助分类或文本生成方法来解决这些问题。另外,在构建连续模板的过程中,需要在预训练模型的基础上添加参数,导致模型在推理时的效率降低。如何微调模型以达到更佳性能,同时只添加少量甚至不添加参数,也是未来的一个研究方向。

5 结语

本文主要总结了自然语言处理中关于提示学习的研究进展。首先描述了预训练语言模型与提示学习的关系,说明现在寻找替代传统微调方法的必要性。然后详细解释了基于提示学习微调模型的步骤,包括提示模板构建、答案搜索和答案映射。接着,本文对提示学习在自然语言处理领域的应用进行举例。最后,对提示学习面临的挑战和未来可能的研究方向进行展望。从研究进展可以看出,提示学习在自然语言处理领域的应用越来越普遍。当有

多个下游任务存在时,用提示学习进行微调可以避免为每一个任务生成一个模型样本,因此是一种轻量化的代替全模型微调策略的方案。虽然提示学习未来的潜力巨大,但是仍然存在许多挑战,例如如何选择最佳的预训练模型、怎样加快训练速度,以及如何将其应用于更多领域都是未来需要研究的方向。

参考文献:

- [1] LIN L D, LIU N, WANG Z A. Adapter and prompt tuning micro-adjustment method research review [J]. Computer Engineering and Applications, 2023, 59(2): 12-21.
林令德, 刘纳, 王正安. Adapter与Prompt Tuning微调方法研究综述[J]. 计算机工程与应用, 2023, 59(2): 12-21.
- [2] LIU P F, YUAN W Z, FU J L, et al. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing [DB/OL]. <http://arxiv.org/abs/2107.13586>.
- [3] KALCHBRENNER N, GREFFENSTETTE E, BLUNSOM P. A convolutional neural network for modelling sentences [C]//Annual Meeting of the Association for Computational Linguistics, 2014: 1-11.
- [4] LIU P F, QIU X P, HUANG X J. Recurrent neural network for text classification with multi-task learning [C]//International Joint Conferences on Artificial Intelligence, 2016: 1-7.
- [5] MARCHEGGIANI D, BASTINGS J, TITOV I. Exploiting semantics in neural machine translation with graph convolutional networks [C]//Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018: 486-492.
- [6] YOUSHA B, AARON C, PASCAL V. Representation learning: a review and new perspectives [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828.
- [7] PETERS M, NEUMANN M, IYYER M, et al. Deep contextualized word representations [C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018: 1-15.
- [8] DEVLIN J, CHANG M, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [DB/OL]. <http://arxiv.org/abs/1810.04805v2>.
- [9] ALEC R, KARTHIK N, TIM S, et al. Improving language understanding by generative pre-training [EB/OL]. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.
- [10] YANG Z, DAI Z, YANG Y, et al. XLNet: generalized auto regressive pre-training for language understanding [C]//Proceedings of the 32nd Annual Conference on Neural Information Processing Systems, 2019: 5754-5764.
- [11] LAN Z, CHEN M, GOODMAN S, et al. ALBERT: a lite BERT for self-supervised learning of language representations [DB/OL]. <https://arxiv.org/abs/1909.11942>.
- [12] CLARK K, LUONG M, LE Q V, et al. Electra: pre-training text encoders as discriminators rather than generators [DB/OL]. <https://arxiv.org/abs/2003.10555>.
- [13] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]// Proceeding of the 31st International Conference on Advances in Neural Information Processing Systems, 2017: 5998-6008.
- [14] YUE Z, ZHANG H, SUN Q, et al. Interventional few-shot learning [C]// 34th Conference on Neural Information Processing Systems, 2020: 2734-2746.
- [15] TIMO S, HINRICH S. It's not just size that matters: small language models are also few-shot learners [C]// Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021: 2339-2352.
- [16] TOM B, BENJAMIN M, NICK R, et al. Language models are few-shot learners [C]// 34th International Conference on Neural Information Processing Systems, 2020, 33: 1877-1901.
- [17] PETRONI F, ROCKTÄSCHEL T, RIEDEL S, et al. Language models as knowledge bases? [C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019: 2463-2473.
- [18] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners [C]// 34th Conference on Neural Information Processing Systems, 2020: 1877-1901.
- [19] SCHICK T, SCHÜTZE H. Exploiting cloze questions for few shot text classification and natural language inference [DB/OL]. <https://arxiv.org/abs/2001.07676>.
- [20] SHIN T, RAZEGHI Y, LOGAN IV R L, et al. Autoprompt: eliciting knowledge from language models with automatically generated prompts [DB/OL]. <https://arxiv.org/abs/2010.15980>.
- [21] GAO T, FISCH A, CHEN D. Making pre-trained language models better few-shot learners [DB/OL]. <https://arxiv.org/abs/2012.15723>.
- [22] RAFFEL C, SHAZEER N, ROBERTS A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer [DB/OL]. <https://arxiv.org/abs/1910.10683>.
- [23] YUAN W, NEUBIG G, LIU P. Bartscore: evaluating generated text as text generation [C]// 35th Conference on Neural Information Processing Systems, 2021: 27263-27277.
- [24] LI X L, LIANG P. Prefix-tuning: optimizing continuous prompts for generation [DB/OL]. <https://arxiv.org/abs/2101.00190>.
- [25] LESTER B, AL-ROUFU R, CONSTANT N. The power of scale for parameter-efficient prompt tuning [C]//2021 Conference on Empirical Methods in Natural Language Processing, 2021: 3045-3059.
- [26] HAMBARDZUMYAN K, KHACHATRIAN H, MAY J. WARP: word-level adversarial reprogramming [C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021: 4921-4933.
- [27] ELSAYED G F, GOODFELLOW I, SOHL-DICKSTEIN J. Adversarial reprogramming of neural networks [DB/OL]. <https://arxiv.org/abs/1806.11146>.
- [28] YIN W P, HAY J, ROTH D. Benchmarking zero-shot text classification: datasets, evaluation and entailment approach [C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019: 3912-3921.
- [29] CHEN X, ZHANG N Y, XIE X, et al. KnowPrompt: knowledge-aware prompt-tuning with synergistic optimization for relation extraction [C]// Proceedings of the ACM Web Conference, 2022: 2778-2788.
- [30] CUI L Y, WU Y, LIU J, et al. Template-based named entity recognition using BART [DB/OL]. <https://arxiv.org/abs/2106.01760>.
- [31] RADFORD A, WU J, SUTSKEVER I, et al. Language models are unsupervised multitask learners [EB/OL]. <https://life-extension.github.io/2020/05/27/GPT%E6%8A%80%E6%9C%AF%E5%88%9D%E6%8E%A2/language-models.pdf>.
- [32] KHASHABI D, MIN S, KHOT T, et al. UnifiedQA: crossing format boundaries with a single QA system [DB/OL] <https://arxiv.org/abs/2005.00700>.