基于 LSTM 和位置增强的软提示向量优化

刘振东,程春玲*,刘 倩
(南京邮电大学 计算机学院,江苏 南京 210023)

摘 要:软提示学习是应用预训练语言模型的新兴方法,然而软提示学习所生成的向量可能缺乏序列结构,影响模型在特定位置定义信息的能力导致模型的性能受损。为此,该文深入探究软提示向量序列结构及其对模型性能的影响,发现软提示向量在不同语言模型类型、模型规模、下游任务类型及提示长度均展现出顺序敏感的问题。针对该问题,提出一种基于 LSTM 和位置增强的软提示排序网络,首先采用改进的 LSTM 网络实现软提示排序调优,其中对每个门控处添设提示选择门,以捕获序列信息生成优序的软提示向量。其次针对排序过程提出一种位置增强模块,结合绝对与相对位置信息优化排序。在 GLUE 数据集上的测试表明,该方法相较于基线带来了平均 3.1% 的性能提升。

关键词:软提示向量;序列结构;顺序敏感性;位置编码;长短期记忆
中图分类号:TP391.1 **文献标识码**:A **文章编号**:1673-629X(2024)10-0118-08
doi:10.20165/j.cnki.ISSN1673-629X.2024.0187

Optimization of Soft Prompt Vectors Based on LSTM and Position Enhancement

LIU Zhen-dong, CHENG Chun-ling*, LIU Qian
(School of Computer Science, Nanjing University of Posts & Telecommunications, Nanjing 210023, China)

Abstract: Soft prompt learning is an emerging method for applying pretrained language models. However, the vectors generated by soft prompt learning may lack sequential structure, affecting the model's ability to define information at specific positions, resulting in impaired model performance. To address this, we delve into the sequential structure of soft prompt vectors and their influence on model performance. It was found that soft prompt vectors exhibit sequence sensitivity issues across different types of language models, model sizes, types of downstream tasks, and prompt lengths. In response, we propose a soft prompt sorting network based on LSTM and position enhancement. Firstly, an improved LSTM network is used for soft prompt sorting optimization, where a prompt selection gate is added at each gate to capture sequence information and generate well-ordered soft prompt vectors. Secondly, a position enhancement module is proposed for the sorting process, optimizing the order by combining absolute and relative position information. Tests on the GLUE dataset show that the proposed method brings an average performance improvement of 3.1% compared to baseline.

Key words: soft prompt vector; sequential structure; order sensitivity; position encoding; long short-term memory

0 引 言

近年来,预训练语言模型在自然语言处理领域大放异彩,显著提升了多种任务的基准性能。在众多方法中,微调[1](Fine Tuning)作为一种主流的应用手段,取得了卓越的成效。然而,随着预训练语言模型参数数量的持续增长,全面微调这些模型变得越来越具挑战性。为应对这一问题,Brown[2]提出了提示学习(Prompt Learning)方法。该方法通过冻结预训练模型,在推理时引入特定的提示信息,以适应不同的下游任务。传统的提示学习需要手工介入。为降低人工设计成本,Lester[3]引入了软提示调优(Soft Prompt Tuning),即在预训练模型输入文本之前加入多个可学习的连续向量,也就是软提示向量,通过梯度传播自动优化这些参数以适应下游任务。软提示学习凭借少量参数的调整,高效应用预训练模型于下游任务,其中软提示向量是关键因素。Lester[3]指出,较长的软提示向量可能因其序列结构问题,使模型难以精准地定位信息。而 Lu[4]发现,在上下文学习中,提示样本的顺序对模型性能有显著影响。

受此启发,该文设计实验对软提示向量序列结构

收稿日期:2024-01-23 **修回日期**:2024-05-28
基金项目:国家自然科学基金项目(61972201)
作者简介:刘振东(1998-),男,硕士研究生,研究方向为自然语言处理;通信作者:程春玲(1972-),女,教授,CCF 会员(15597M),研究方向为数据挖掘、数据管理。

进行研究,实验发现软提示向量存在特定的序列结构,并具有顺序敏感性。在不同模型、模型规模和提示长度下都会影响模型预测性能。为了寻得具有较优序列的软提示,提出一种基于 LSTM 和位置增强的软提示向量优化方法,通过软提示元素顺序调优实现软提示优化。该文的主要贡献包括:

(1)发现软提示向量在不同模型、模型规模、下游任务、提示长度上均存在序列结构,且该序列结构具有顺序敏感性。

(2)提出了一种软提示排序网络提取软提示向量中的时序信息,重新排列寻找软提示中较优的元素顺序,实现软提示向量调优。

(3)针对软提示向量序列结构,提出了一种位置增强模块,为软提示向量添加位置信息,辅助软提示排序调优。

1　相关工作

提示学习是当前应用大规模预训练语言模型的主流方法,它的核心思想是在模型输入中加入某种形式的提示信息,以指导模型更好地理解和处理特定的任务。提示学习通过提示工程在下游任务上找到合适的提示[5],提示工程主要分为人工提示工程和自动提示工程。

人工提示工程依赖于人工经验来寻找最合适的提示模板。LAMA 数据集[6]通过人工创建的填空提示来挖掘语言模型中的知识。Brown[2]手工设计前缀提示,以处理包括问答、翻译和常识推理探测任务在内的多种任务。Schick 和 Schütze[7-8]在文本分类和条件文本生成任务中使用了人工预定义的提示进行少样本学习。尽管人工提示工程是简单直接的,但对于一些复杂的任务,即使是经验丰富的提示设计师也可能无法找到最佳提示[9]。

自动提示工程则通过让模型自己寻找高质量提示来进行提示学习。自动提示工程根据提示类型的不同分为离散提示和连续提示。离散提示是在离散空间中自动搜索自然语言文本构成的提示。Jiang[9]通过挖掘算法找到输入与输出之间的中间联系词构建提示。Wall[10]则提出以某一提示词作为起点进行梯度搜索来找到起点附近的自然语言文本作为提示。Gao[11]则尝试通过算法让预训练模型直接生成下游任务的相关提示。

不同于由自然语言文本构成的离散提示,连续提示由连续空间中的一组向量构成,连续提示又被称为软提示。Lester[3]通过在模型输入前设置多个可学习的连续向量,优化了离散提示的限制,拓展了提示所能表征的特征空间,同时避免了人工设计提示的成本。软提示向量生成方式是目前研究的重点,Vu[12]尝试通过将多个源任务知识迁移到目标任务上,将多个源任务生成的软提示向量作为目标任务的初始化起点。Zhong[13]则利用知识蒸馏技术有效地从源任务提示传递知识到目标任务提示,而 Asai[14]则尝试在多个源任务知识迁移过程中,将源任务提示向量与目标任务提示进行交互,使目标提示向量更加关注源提示向量中重要的内容。软提示向量自身的特性也值得研究,而目前的相关研究较为匮乏。Yang[15]尝试通过任务动态地调整提示的内部与外部因素,例如提示长短。尽管 Lester[3]关注到软提示向量内部的序列结构可能使模型难以将信息定义到特定的位置,但其未展开进一步研究。

2　软提示顺序敏感性分析

自然语言文本构成的提示通常具有一定的序列结构,这种语序包含着上下文信息,影响模型的预测。以“今天的天气很坏,我的心情却很好”为例,若将“很坏”和“很好”互换位置,句子的整体含义就会发生根本变化。受此启发,对于由连续向量构成的软提示,该文设计多个实验,从更换顺序的方式、预训练模型的种类与大小、软提示的不同长度等多个角度观察改变软提示向量中元素之间的顺序对模型预测性能的影响,探究其内部的序列结构。该文首先在 Bert[16]模型通过不同更换顺序方式探究软提示序列结构。随后,更换模型种类与大小、不同的下游任务进一步探究软提示相关的序列结构特性。

2.1　软提示向量的顺序敏感性

首先,本研究选取 GLUE[17]数据集中的 SST-2[17]任务作为下游任务来探究在 Bert-base[16]和 Bert-large[16]模型上软提示的序列结构。具体而言,在这两个模型上通过软提示学习方法生成了长度为 100 的软提示向量,每个向量由 100 个元素组成。随后,对这些学习到的软提示向量进行了实验:以单个向量元素为单位,随机交换元素之间的位置,从而构造出新的软提示向量。在 Bert-base 和 Bert-large 模型上,分别生成了 500 个具有不同元素顺序的软提示向量,并观察其性能表现,评估指标为准确率。

表 1 中较差、较优、基准性能数分别代表换序后预测性能相较于基准降低、提高、不变的软提示顺序个数。如表 1 所示,大多数重新排序的软提示向量在下游任务的性能表现都发生了变化。具体来说,在 Bert-base 和 Bert-large 模型上换序得到的 500 个具有新顺序的软提示向量中,分别有 98.6% 和 97.8% 的软提示向量性能出现下降;0.6% 和 1% 的软提示向量性能保持不变;仅有 0.8% 和 1.2% 的软提示向量相比基准性

能有所提高。从实验结果来看,在 Bert-base 和 Bert-large 模型上,相较于不换序的基准性能,存在换序的软提示向量导致模型性能下降 1.4 百分点和 4.3 百分点。这种现象与自然语言文本中改变句子语序影响模型预测性能的现象类似,而这正是具有序列结构的表现。为了验证这一结论有效性,该文尝试在软提示向量作用于模型时,不为软提示向量引入位置向量。在不考虑序列元素位置的模型如 transformer 中,位置向量通过为每个元素附加位置信息,帮助模型理解序列中元素的顺序。而不为文本添加位置向量会限制模型的性能,因为它忽略了文本的序列结构。如果输入本身并不具有序列结构,那么去除位置向量将不会对输入产生影响。实验结果如表 1 所示,在 Bert-base 和 Bert-large 模型上不为软提示引入位置向量相较于引入位置向量的基准性能,模型性能分别下降了 1 百分点和 0.7 百分点。说明模型缺失了对软提示序列的关注。这也侧面证明了软提示向量是拥有序列结构的一种输入。将改变软提示元素顺序影响模型性能这种现象称为“软提示顺序敏感性”。同时,在 Bert-base 和 Bert-large 模型实验中,分别观察到了 0.1 百分点和 0.8 百分点的性能提升,这表明通过调整生成的软提示向量的顺序,可以对其进行优化,从而提高模型性能。

为了进一步探究不同预训练模型对软提示向量序列结构敏感性的影响,在 T5-base[1] 和 RoBERTa-base[18] 模型上复制了之前在 Bert 模型上进行的实验。具体而言,在这两个模型上学习了长度为 100 的 SST-2 任务软提示向量,并随机交换单个元素位置,以生成各自的 500 个不同顺序的软提示向量,如表 1 所示。

表 1 不同模型在 SST-2 任务软提示换序结果

模型	不换序	去位置向量	换序				
	基准性能/%	基准性能/%	最差性能/%	最佳性能/%	较差顺序数	较优顺序数	基准性能数
Bert-base	89.9	88.9	88.5	91.0	493	4	3
Bert-large	92.3	91.6	88.0	93.1	489	6	5
T5-base	92.3	92.1	91.6	94.3	487	6	7
RoBERTa-base	93.1	92.8	92.2	93.8	474	16	10

在 T5-base 和 RoBERTa-base 模型上,新顺序的软提示向量同样影响了模型预测性能:在 T5-base 模型上观察到 2 百分点的性能提升,而在 RoBERTa-base 模型上观察到 0.7 百分点的提升。同时去位置向量后,T5-base 和 RoBERTa-base 模型的性能分别下降了 0.2 百分点和 0.3 百分点,这表明序列结构是多个预训练模型学习到的软提示普遍具有的,并且软提示顺序敏感性也是在不同预训练模型上普遍存在的。

为了深入探索软提示序列结构对模型预测性能的影响,本研究采用了不同的换序方法。换序方式 1 涉及将向量元素按固定长度分组,并以这些组为单位进行交换,分别尝试了 5,10,20 个元素为一组的设置。换序方式 2 则固定部分元素位置不变,仅交换其他元素的位置,例如固定前 80 个元素不变,只交换后 20 个元素。在这些实验设置下,生成了 200 个新顺序的软提示向量,以评估不同换序方式对模型性能的影响。需要注意的是,在一定长度的组合下,可能性有限,因此实验次数将基于该设置下的最大可能组合数进行。换序方式 1 实验结果,如表 2 所示。

表 2 的实验结果表明,在不同长度的交换基本单位下,模型预测性能均受到软提示向量内部元素顺序的显著影响。特别地,当交换基础单位为单个元素时,性能波动的幅度最大,达到了 5.2 百分点。随着交换基本单位长度的增加,观察到模型预测性能的波动范围逐渐减小。换序单位长度为 20 相较于单位长度为 1 最差性能提升了 2.3 百分点,这说明软提示内部存在一定元素聚集形成序列,这使得换序的结果更加稳定,但较长的换序单位长度也限制了换序调整软提示向量的性能的上限,在单位长度最小时软提示换序实现了最佳性 93.1%,相较于单位长度为 20 时提升了 0.3 百分点。这一现象揭示了软提示向量内部元素按照一定长度聚集成序列结构的特点,软提示向量不仅仅是独立元素的简单组合,而是具有一定的内在结构特性,这对模型的预测性能产生重要影响。

表 2 SST-2 任务软提示换序方式 1 结果

单位元素长度	换序				
	最差性能/%	最佳性能/%	较差顺序数	较优顺序数	基准性能数
1	87.9	93.1	191	6	3
5	89.4	92.7	194	4	2
10	89.8	92.8	190	4	6
20	90.2	92.8	191	7	2

如表 3 所示,在软提示向量中固定部分元素位置不变时,模型的预测性能表现出了显著的改善。具体来说,当固定向量中前 90 个元素位置时,较优顺序的占比高达 54%;而在固定后 90 个元素位置时,这一比

例更是提升至 76%。这一结果明显优于先前实验中采用的其它换序方式。该文认为这是因为学习到的软提示向量本身就具有一定的序列结构,针对小范围的序列结构调整对于整体的序列结构破坏性更小,同时保存了学习到的部分序列结构信息,从而更易挖掘出更加优秀的序列结构,而在固定元素数量较少时,更多的序列结构被重新构建,有可能导致先前的较优序列结构被破坏,从而使换序结果更佳不稳定,出现了较多的相较于基准性能下降的软提示顺序。这一发现表明软提示向量中存在序列聚集和特定的顺序结构。

表 3　SST-2 任务软提示换序方式 2 结果

固定元素下标	换序				
	最差性能/%	最佳性能/%	较差顺序数	较优顺序数	基准性能数
0-50	91.0	93.1	133	43	14
0-80	91.5	92.9	97	71	32
0-90	91.7	92.9	58	108	34
50-99	91.2	92.7	156	19	25
20-99	91.8	92.8	82	76	42
10-99	92.2	92.8	1	152	47

鉴于 T5 模型在软提示应用中的通用性和有效性,本研究后续实验将主要以 T5 模型为基准。同时,考虑到单位长度为 1 的随机换序方式在性能浮动范围上表现最为显著,将采用此换序方法继续深入探索软提示向量的序列结构特性。

2.2　软提示顺序敏感性与下游任务

为了研究下游任务对软提示顺序敏感性的影响,该文在 T5-base 模型上分别对 STS-B[17]、RTE[17] 和 MRPC[17] 三个标准数据集复制了表 1 中的实验,以探究下游任务与软提示向量顺序敏感性之间的关系。实验结果如表 4 所示。从表 4 的实验结果来看,在 STS-B、RTE 和 MRPC 三个任务中,通过换序操作的软提示向量均导致模型性能的波动。

表 4　不同下游任务软提示换序结果

数据集	评价指标	换序		
		基准性能	最差性能	最佳性能
STS-B	Pr/%	90.3	90.2	91.0
RTE	Acc/%	54.7	44.6	61.1
MRPC	Acc/%	68.1	27.9	70.6

其中,MPRC 为释义检测任务,STS-B 为句间相似性检测任务,尽管是不同的任务类型,不同的任务评价指标,软提示元素顺序的改变都显示出了模型性能的浮动,而这种浮动正是序列结构存在的表现,这些结果显示出在不同下游任务中软提示向量普遍存在顺序敏感性。特别是在 MRPC 任务中,改变元素顺序导致的模型预测性能下降幅度高达 40.2 百分点,而在 RTE 任务中,换序操作却带来了 6.4 百分点的性能提升。不同下游任务浮动范围的上下限并不相同,表明软提示向量顺序优化潜力在一定程度上取决于下游任务。

2.3　软提示顺序敏感性与提示长度

为了进一步探究软提示长度与顺序敏感性之间的关系,本研究选择了具有适中数据集大小的 STS-B 任务作为下游任务。在这一实验中,基于 T5-base 模型,针对 STS-B 任务学习了不同长度(5,10,20,50,100)的软提示。随后,对这些不同长度的软提示向量复制了表 1 的实验,所有实验结果的评估指标均采用皮尔逊相关系数[17](Pearson correlation coefficient)。

从表 5 中可以看到,在提示长度较短的 5,模型性能因为提示元素顺序改变最低下降了 0.1 百分点,最高提升了 0.4 百分点。而对于较长的 100 时,同样模型性能最低降了 0.1 百分点,最高提升了 0.7 百分点,在不同长度下软提示向量元素的顺序的改变均显著影响了模型的预测性能。证明了在不同的提示长度下,软提示向量均存在一定的序列结构,会因为元素顺序改变被破坏,从而影响模型预测性能。同样的,在长度分别为 10 和 20 的向量中,这种波动分别达到了 0.5 百分点和 0.6 百分点。即使这种较短的软提示向量,也会因为改变其元素顺序影响其序列结构,因此对于较短的软提示向量其内部元素的顺序也不容忽视,这种软提示顺序敏感性是在不同提示长度下普遍存在的。

表 5　不同长度软提示换序结果

提示长度	换序		
	基准性能	最差性能	最佳性能
5	90.3	90.2	90.7
10	90.6	90.5	91.0
20	90.5	90.3	90.9
50	90.3	90.0	90.8
100	90.3	90.2	91.0

2.4　软提示顺序敏感性与模型规模

模型规模也可能是影响软提示序列结构的因素之一,软提示顺序敏感性也可能随模型规模变化,因此该文在 T5-small、base、large 模型规模下学习得到 STS-B 任务生成长度为 100 的软提示,并对学习到的软提示元素进行随机换序,得到 500 个不同顺序的软提示向量,观察其对模型性能的影响,实验结果如表 6 所示。

从表 6 的实验结果来看,更换软提示顺序在 small 模型规模上产生性能下降为 0.9 百分点,而在 base 与 large 模型上分别产生了 0.1 百分点与 1.7 百分点的性

能下降。可以看到这种改变软提示元素顺序对于模型影响是跨模型规模存在的,说明在不同模型规模上学习到的软提示向量均存在序列结构。同时,在模型规模 base 上产生性能提升 0.7 百分点,而随着模型规模增加到 large 性能提升了 0.5 百分点与 small 提升浮动持平。可以看出软提示序列结构对模型的影响并不因模型大小产生明显区别。

表 6 不同模型规模软提示换序结果

模型规模	换序		
	基准性能	最差性能	最佳性能
T5-small	87.8	86.9	88.3
T5-base	90.3	90.2	91.0
T5-large	91.2	89.5	91.7

综上所述,可以看出:

(1)软提示向量具有一定的序列结构,同时软提示向量存在顺序敏感性。

(2)在不同模型上学习到的软提示向量都表现出顺序敏感性。顺序敏感性跨下游任务、软提示长度、模型规模存在。

3 基于 LSTM 和位置增强的软提示向量优化模型

由于软提示向量对向量中元素的顺序敏感,如果能够捕捉和利用这些向量中的序列信息来优化其元素的顺序结构,可以降低这种顺序敏感性,提高软提示学习的性能。针对软提示向量的顺序敏感性,基于 LSTM 网络提出一种软提示向量排序优化网络,通过对软提示向量元素重新排序对提示向量寻优,提高软提示方法在下游任务上的性能。

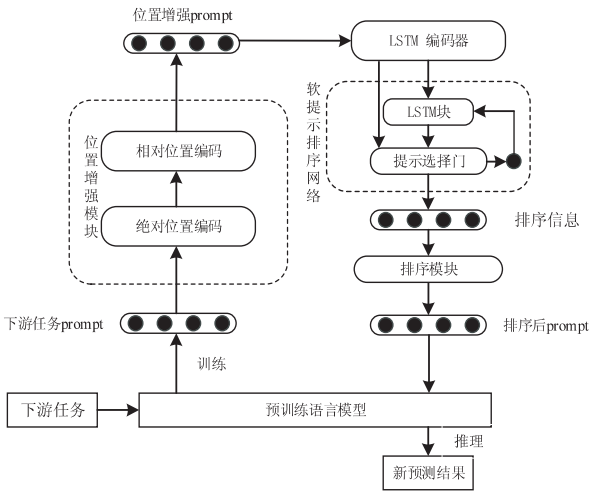

图 1 软提示排序框架

模型结构如图 1 所示,主要包括两个部分,第一个部分为位置增强模块,用于为软提示向量增添序列信息,改善其顺序结构。第二部分是基于 LSTM 架构的软提示排序网络,首先通过编码器提取软提示向量中的时序信息,随后通过解码器中提示选择门进一步加深序列信息之间的交互,同时将编解码器提取到的时序信息转化为排序模块所需的排序信息,最终排序模块对提示选择门提供的信息进行编码与梯度软化实现重新排序。

3.1 位置增强模块

在注意力网络中,位置编码是为文本信息提供空间上联系的常用技术。而软提示向量具有顺序敏感的特性,该文借鉴这一技术,通过向软提示向量增添位置信息来助力序列信息的有效提取,为软提示排序网络提供辅助。如图 2 所示,该模块主要由两部分构成:绝对位置增强部分和相对位置编码部分。

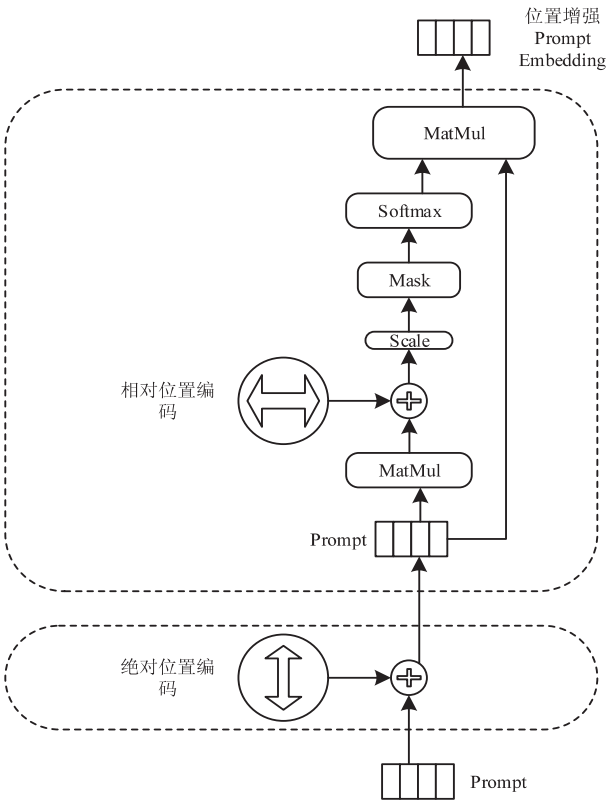

图 2 位置增强模块框架

绝对位置增强部分采用基于三角正弦的位置向量对软提示向量进行绝对位置编码:

$$P_a = P + \mathrm{PE}_a \tag{1}$$

$$\mathrm{PE}_{\mathrm{pos},2i} = \sin\left(\frac{\mathrm{pos}}{10\,000^{\frac{2i}{d_m}}}\right) \tag{2}$$

$$\mathrm{PE}_{\mathrm{pos},2i+1} = \cos\left(\frac{\mathrm{pos}}{10\,000^{\frac{2i+1}{d_m}}}\right) \tag{3}$$

其中,PE_a 为绝对位置矩阵,i 代表着时间步,d_m 代表着软提示特征维度。该过程使排序网络能够对软提示向量元素之间的绝对位置信息给予更多的关注。

同时应用相对位置编码来强化软提示向量元素间的相对位置联系。过程如下:

$$Q,K,V = \mathrm{Linear}(P_a) \tag{4}$$

$$\mathrm{Att}_{\mathrm{score}} = \mathrm{MatMul}(Q,K) + \mathrm{MatMul}(Q,\mathrm{relative}_{\mathrm{emb}}) \tag{5}$$

$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{6}$$

$$P_{\mathrm{POS}} = \mathrm{Concat}(\mathrm{head}_1,\cdots,\mathrm{head}_{\mathrm{num}_{\mathrm{head}}})W^O \tag{7}$$

其中, MatMul 代表矩阵乘法, $\mathrm{relative}_{\mathrm{emb}}$ 为相对位置嵌入, Concat 为拼接操作。该过程为软提示添加相对位置特征。最终得到软提示向量 P_{POS} 帮助软提示排序网络进行排序操作。

3.2　编解码器与提示选择门

编码器选择 LSTM 网络作为基础架构,利用 LSTM 网络有效处理和提取长序列信息的能力得到软提示向量的编码向量 $\mathrm{En}_{\mathrm{out}}$ 与 $\mathrm{En}_{\mathrm{hid}}$,二者包含了软提示向量内部的序列信息。随后将 $\mathrm{En}_{\mathrm{out}}$ 和 $\mathrm{En}_{\mathrm{hid}}$ 作为解码器第一个时间步的输入。这样的设计是为了确保软提示向量编码后的信息能与解码器紧密结合,以此加深软提示向量中序列信息的交互。

解码器与提示选择门网络框架如图 3 所示。

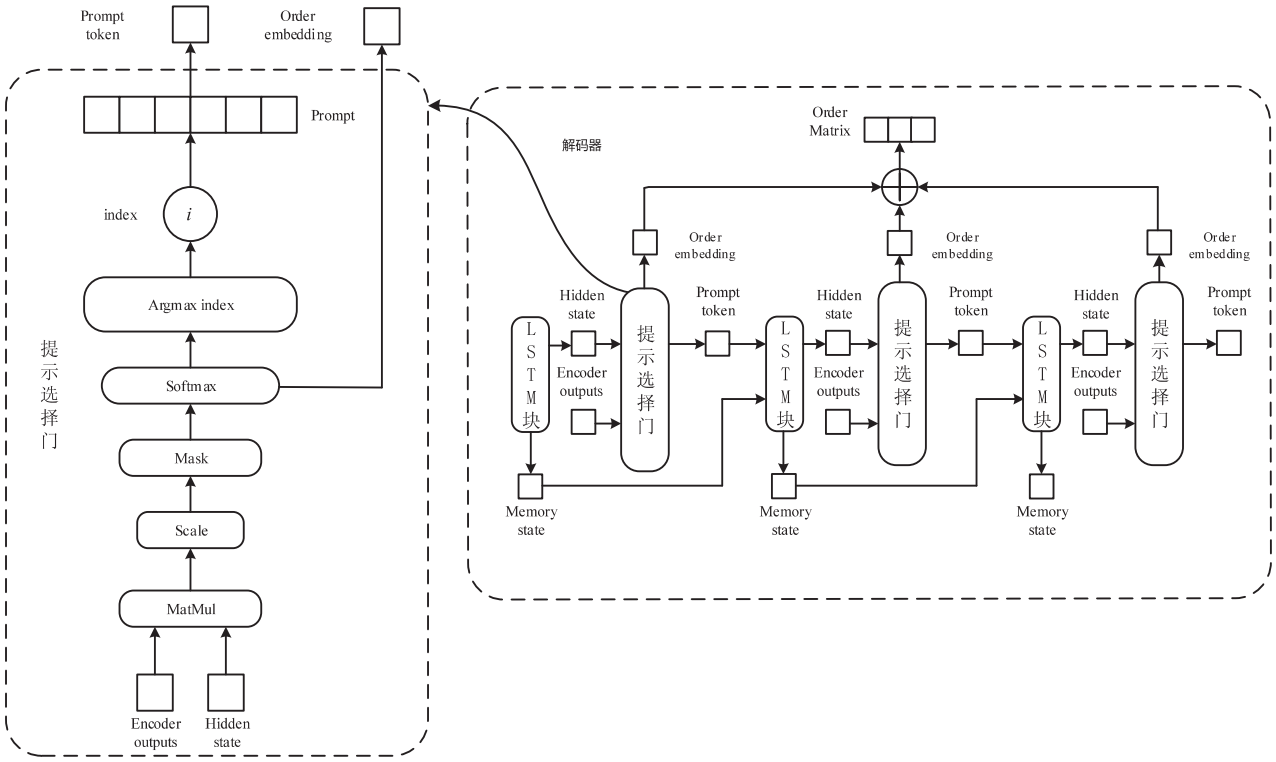

图 3　解码器与提示选择门网络框架

因为 LSTM 可以在较长时间跨度内保持信息,首先由 LSTM 单元输出包含当前时间步时序信息的隐藏状态 h_i :

$$h_i = \mathrm{LSTM}(h_{i-1},p_{i-1},m_{i-1}) \tag{8}$$

其中, p_{i-1} 代表该时间步被选择的软提示元素, m_{i-1} 为记忆信息。

为了能进一步增强模型在每个时间步的选择性注意,以达到更加有效的时序信息提取,该文提出了一种提示选择门,提示选择门中的流程如下:

$$V_i = \mathrm{Attention}(\mathrm{En}_{\mathrm{out}},h_i) \tag{9}$$

$$O_i = \mathrm{Softmax}(V_i) \tag{10}$$

$$p_i = j^* = \mathrm{Argmax}(O_i) \tag{11}$$

其中, O_i 中每个元素表征了第 i 个时间步与该元素相同索引的软提示向量元素被选择的概率。j^* 代表当前时间步被选择概率最高元素的索引, p_i 为该索引所对应的软提示元素。被选择的软提示元素加深了软提示元素之间的信息交互,实现了时序信息的进一步提取与传递。

3.3　排序模块

该文利用解码器中生成的包含着时序选择信息的向量 O_i ,实现对软提示向量元素的重新排序,排序过程如下:

$$\mathrm{Order} = \mathrm{Concat}(O_1O_2\cdots O_n) \tag{12}$$

$$O_h = \mathrm{Onehot}(\mathrm{Argmax}(\mathrm{Order})) \tag{13}$$

$$O_h = O_h + \mathrm{Order}_{\mathrm{detach}} - \mathrm{Order} \tag{14}$$

$$P_{\mathrm{order}} = \mathrm{MatMul}(O_h,P) \tag{15}$$

其中, Order 由 O_i 拼接得到。Argmax 选择出对应索引,索引与原软提示向量中的元素一一对应,通过索引选择软提示元素实现重新排序。但由于位置索引是离散的,无法应用反向传播算法更新网络参数,该文使用了梯度软化技术,对索引应用独热编码得到 O_h ,独热编码的目的是为了保证离散的索引可以连续化。随后对 Order 矩阵进行复制并剥离其梯度。这一步骤确保了在正向推理过程中的离散性以及在反向传播过程中的

柔性分布。最终得到排序后软提示 P_{order}。

4 实验结果与分析

4.1 数据集与实验设置

通过 GLUE[17] 数据集中的 7 个子任务来评估文中方法的效果。数据集主要涉及情感分析、释义检测、蕴含等任务,包括 RTE,STS-B,MRPC,MNLI,QQP,QNLI 和 SST-2。各任务划分如表 7 所示。

表 7 实验数据集划分

	训练集	验证集	测试集	总数	任务类型
RTE	2.49 k	0.27 k	3 k	5.77 k	蕴含
MRPC	3.67 k	0.40 k	1.73 k	5.8 k	释义检测
STS-B	5.75 k	1.5 k	1.38 k	8.63 k	句间相似
SST-2	67.3 k	0.87 k	1.82 k	70 k	情感分类
QNLI	105 k	5.46 k	5.46 k	116 k	蕴含
QQP	364 k	40.4 k	391 k	795 k	语义等价
MNLI	393 k	19.6 k	19.6 k	432 k	蕴含

以微调[1](Fine Tuning,FT)和软提示调优[3](Soft Prompt Tuning,PT)方法作为对比基线,使用 T5-base 模型为基准模型,FT 和 PT 方法依照 Lester[3] 的方法复现作为对比,在上述任务测试集上与文中方法进行性能比较,性能基准除了 STS-B 任务为皮尔逊相关系数外,其它性能基准均为准确率。

模型的实现基于 Pytorch 框架。使用 T5-base 模型,从 huggingface 库中加载预训练参数初始化模型。训练软提示和排序网络的过程中,设置 batch_size 为 16 或 32,在{1e-1,5e-5}范围内对学习率进行搜索。模型使用 Adam 优化器。训练提示向量的 epoch 数设置为 5,10 或 20。

4.2 实验结果与分析

首先在不同任务上通过软提示学习得到各个任务长度为 100 的软提示向量。随后将每个任务得到的软提示向量送入文中提出的位置增强模块与软提示排序网络中,在冻结软提示向量参数的前提下,得到新顺序的软提示向量送入预训练模型中,以此更新排序网络的参数。在下游任务上学习单个 epoch 后对经过排序的软提示向量在下游任务上的性能进行评估,结果如表 8 所示。

表 8 不同方法在 GLUE 数据集上的实验结果 %

对比方法	任务							
	MNLI	QQP	QNLI	SST-2	STS-B	MRPC	RTE	Avg.
	Acc	Acc	Acc	Acc	Pr	Acc	Acc	Acc
FT	86.8	91.6	93.0	94.6	89.7	90.2	71.9	88.2
PT	84.2	89.7	92.8	90.9	90.3	68.1	54.7	81.5
Ours	84.5	90.3	93.0	91.5	90.6	69.6	59.0	82.6
Ours+位置增强	84.5	90.3	93.1	91.9	91.0	72.1	59.7	84.6

文中方法帮助软提示学习在 GLUE 数据集的各个子任务上带来了明显的性能提升。具体而言,文中方法相较于传统软提示学习 PT 方法在 GLUE 的整体任务上平均提高了约 1.1 百分点,尤其在 RTE 任务上实现了 4.3 百分点的显著性能提升。这一成效可归因于本研究采用的 LSTM 网络有效提取了软提示向量中的上下文信息,解码器中的提示选择门进一步捕获了软提示元素间的序列信息,并通过该选择门对软提示向量进行了顺序调整,有效缓解了软提示顺序敏感的问题,而与 FT 方法相比,文中方法仍存在差距。这是因为 FT 针对特定的任务可以调整模型的全部参数,挖掘出更深层次的特征,微调仍是多数特定下游任务的 SOTA 性能基线[3]。尽管文中方法在性能上仍与微调存在差距,但文中方法保持了 PT 方法的存储成本小的优点和模型的通用性。在 QQP、SST-2 等任务上更是缩小了 PT 与 FT 之间性能的差距(0.6 百分点与 1.0 百分点),同时在 QNLI 任务上使 PT 甚至超过了 FT 方法 0.1 百分点。在保持冻结模型的基础上,仅通过排序网络找到更优的软提示向量顺序,也避免了微调模型和更新软提示向量参数,这也是文中方法相较于 FT 的优势所在,并且为软提示的应用提供了更好的基础。

而经过位置增强的软提示排序方法相较于传统软提示学习 PT 方法,在性能上带来了接近平均 3.1 百分点的提升。尤其在在初始性能一般的任务上带来了显著的性能提升。例如,在 RTE 任务中,通过软提示学习得到的准确率为 54.7%,而经过文中方法优化后提升至 59.7%,增幅达 5 百分点。同样,在 MRPC 任务中,性能从 68.1% 提高至 72.1%,增加了 4 百分点。同时,与微调 FT 方法的性能差距也得到了显著缩小,特别是在 STS-B 任务上,文中方法甚至超越了微调方法的基线。这表明软提示顺序敏感性为这些任务提供了显著的提升空间,而顺序优化策略有效挖掘了这一潜力,从而显著提升了基线性能。相比之下,对于已有较高性能的任务如 QNLI(已达 92.8% 准确率),文中方法仅带来了 0.3 百分点的提升。虽然增益相对较小,但仍证明了该方法的有效性和普遍适用性,体现出

位置增强模块为软提示排序网络增添了关键的位置信息。

另外还发现,随着数据集的减少,文中方法带来的性能提升逐渐增加。在大型数据集如 MNLI、QQP、QNLI 上,性能提升不足 1 百分点;而在 MRPC、RTE 这样的小型数据集上,提升达到了 4 百分点至 5 百分点。这说明在小数据集上,调整软提示的顺序能带来更显著的影响,因为每个提示在这些情况下变得更为关键。文中方法使模型更加注重提示之间的顺序,使小数据集上软提示的泛化性得到了提高。

5　结束语

探究了软提示向量的序列结构,发现软提示向量序列的顺序敏感性。该现象跨模型、任务、模型规模、提示长度存在。提出的软提示排序网络和位置增强模块利用其序列结构为软提示向量找到了较优的元素排列顺序,提高了软提示学习的性能。

参考文献:

[1] RAFFEL C, SHAZEER N, ROBERTS A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. The Journal of Machine Learning Research, 2020, 21(1): 5485-5551.

[2] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020, 33: 1877-1901.

[3] LESTER B, AL-RFOU R, CONSTANT N. The power of scale for parameter-efficient prompt tuning[C]//Proceedings of the 2021 conference on empirical methods in natural language processing. Stroudsburg: ACL, 2021: 3045-3059.

[4] LU Y, BARTOLO M, MOORE A, et al. Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity[C]//Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers). Stroudsburg: ACL, 2022: 8086-8098.

[5] LIU P, YUAN W, FU J, et al. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing[J]. ACM Computing Surveys, 2023, 55(9): 1-35.

[6] 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 (EMNLP-IJCNLP). Stroudsburg: ACL, 2019: 2463-2473.

[7] SCHICK T, SCHÜTZE H. Few-shot text generation with pattern-exploiting training[J]. arXiv: 2012.11926, 2020.

[8] SCHICK T, SCHÜTZE H. Exploiting cloze-questions for few-shot text classification and natural language inference[C]//Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume. Online: ACL, 2021: 255-269.

[9] JIANG Z, XU F F, ARAKI J, et al. How can we know what language models know? [J]. Transactions of the Association for Computational Linguistics, 2020, 8: 423-438.

[10] WALLACE E, FENG S, KANDPAL N, et al. Universal adversarial triggers for attacking and analyzing NLP[C]//Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). Stroudsburg: ACL, 2019: 2153-2162.

[11] GAO T, FISCH A, CHEN D. Making pre-trained language models better few-shot learners[C]//Joint conference of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing, proc of the annual meeting on association for computational linguistics. Stroudsburg: ACL, 2021: 3816-3830.

[12] VU T, LESTER B, CONSTANT N, et al. SPoT: better frozen model adaptation through soft prompt transfer[C]//Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers). Stroudsburg: ACL, 2022: 5039-5059.

[13] ZHONG Q, DING L, LIU J, et al. Panda: prompt transfer meets knowledge distillation for efficient model adaptation[J]. arXiv: 2208.10160, 2022.

[14] ASAI A, SALEHI M, PETERS M E, et al. Attempt: parameter-efficient multi-task tuning via attentional mixtures of soft prompts[C]//Proceedings of the 2022 conference on empirical methods in natural language processing. Stroudsburg: ACL, 2022: 6655-6672.

[15] YANG X, CHENG W, ZHAO X, et al. Dynamic prompting: a unified framework for prompt tuning[J]. arXiv: 2303.02909, 2023.

[16] KENTON J D M W C, TOUTANOVA L K. BERT: pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of NAACL-HLT. Stroudsburg: ACL, 2019: 4171-4186.

[17] WANG A, SINGH A, MICHAEL J, et al. GLUE: a multi-task benchmark and analysis platform for natural language understanding[C]//Proceedings of the 2018 EMNLP workshop BlackboxNLP: analyzing and interpreting neural networks for NLP. Stroudsburg: ACL, 2018: 353-355.

[18] LIU Zhuang, LIN Wayne, SHI Ya, et al. A robustly optimized BERT pre-training approach with post-training[C]//Proceedings of the 20th Chinese national conference on computational linguistics. Huhhot: Chinese Information Processing Society of China, 2021: 1218-1227.