北京大学学报(自然科学版) 第 60 卷 第 1 期 2024 年 1 月
Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 60, No. 1 (Jan. 2024)
doi: 10.13209/j.0479-8023.2023.071

增强提示学习的少样本文本分类方法

李睿凡[1,2,3,†] 魏志宇[1] 范元涛[1] 叶书勤[1] 张光卫[2,4]

1. 北京邮电大学人工智能学院, 北京 100876; 2. 教育部信息网络工程研究中心, 北京 100876; 3. 交互技术与体验系统文化和旅游部重点实验室, 北京 100876; 4. 北京邮电大学计算机学院, 北京 100876; †E-mail: rfli@bupt.edu.cn

摘要 针对少样本文本分类任务, 提出提示学习增强的分类算法(EPL4FTC)。该算法将文本分类任务转换成基于自然语言推理的提示学习形式, 在利用预训练语言模型先验知识的基础上实现隐式数据增强, 并通过两种粒度的损失进行优化。为捕获下游任务中含有的类别信息, 采用三元组损失联合优化方法, 并引入掩码语言模型任务作为正则项, 提升模型的泛化能力。在公开的 4 个中文文本和 3 个英文文本分类数据集上进行实验评估, 结果表明 EPL4FTC 方法的准确度明显优于所对比的基线方法。

关键词 预训练语言模型; 少样本学习; 文本分类; 提示学习; 三元组损失

Enhanced Prompt Learning for Few-shot Text Classification Method

LI Ruifan[1,2,3,†], WEI Zhiyu[1], FAN Yuantao[1], YE Shuqin[1], ZHANG Guangwei[2,4]

1. School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876; 2. Engineering Research Center of Information Networks, Ministry of Education, Beijing 100876; 3. Key Laboratory of Interactive Technology and Experience System, Ministry of Culture and Tourism, Beijing 100876; 4. School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876; † E-mail: rfli@bupt.edu.cn

Abstract An enhanced prompt learning method (EPL4FTC) for few-shot text classification task is proposed. This algorithm first converts the text classification task into the form of prompt learning based on natural language inference. Thus, the implicit data enhancement is achieved based on the prior knowledge of pre-training language models and the algorithm is optimized by two losses with different granularities. Moreover, to capture the category information of specific downstream tasks, the triple loss is used for joint optimization. The masked-language model is incorporated as a regularizer to improve the generalization ability. Through the evaluation on four Chinese and three English text classification datasets, the experimental results show that the classification accuracy of the proposed EPL4FTC is significantly better than the other compared baselines.

Key words pre-trained language model; few-shot learning; text classification; prompt learning; triplet loss

文本分类[1]是自然语言处理领域的热点研究内容之一, 已经在多个场景得到充分的发展。例如, 在情感分析、新闻推荐和用户画像等场景中, 通常可以获取海量的未标注数据, 因而需要大量的人工标注工作。但是, 在诸如医疗和安全等一些特殊的工业应用场景中, 通常较难获取大量数据来满足模型的训练, 使得基于数据驱动的深度学习方法在少量数据情况下较难取得令人满意的效果。为了使机器具有仅通过几个简单样本就能实现快速学习新事物的能力, Wang 等[2]提出少样本学习的概念。少样本学习的核心目标是面对新的领域任务, 利用先验知识, 仅通过有限的训练样本, 快速且准确地完成对新领域任务的学习。

近年来, 随着预训练语言模型的发展, 尤其是以 BERT[3]为代表的通用预训练模型的提出, 使得基于预训练和微调的两阶段训练范式逐渐成为新的趋势, 并在大多数自然语言处理任务中取得优异的成绩。但是, 在微调阶段, 模型的性能通常取决于

国家自然科学基金(62076032)资助
收稿日期: 2023–05–18; 修回日期: 2023–08–30

任务的类型和有标注训练数据的规模。这就使得当模型面对仅含有少量训练样本的下游任务时，往往性能表现不佳。针对预训练模型下的少样本学习问题，基于提示学习的方法提供了一种新颖有效的解决思路。基于提示学习的方法通过将下游任务的形式调整为与预训练任务形式一致，充分发挥预训练模型中语言模型任务的优势，同时通过减小上下游任务训练方式不一致带来的差异，达到少样本学习的目的。虽然基于提示学习的方法已取得不错的效果，但仍然面临以下两方面的挑战：1) 在少样本学习的场景中，容易出现类别的数量远多于单一类别样本量的现象，使得模型在此类任务中的表现通常较差；2) 基于提示学习的方法大多依赖预训练语言模型中已经学习到的先验知识，较少关注下游任务的类别表征信息。

针对上述问题，本文提出一种增强提示学习的少样本文本分类算法 EPL4FTC (enhanced prompt learning for few-shot text classification)。该算法首先将下游任务转换成基于自然语言推理的提示学习形式，通过任务形式的转换，在有效地利用预训练语言模型中已学习到的先验知识的基础上，实现数据的隐式增强，并通过两种粒度的损失进行优化。此外，为捕获下游任务中丰富的类别表征信息，该算法通过三元组损失[4]进行联合优化，同时引入掩码语言模型任务(MLM)作为正则项，预防过拟合或数据灾难性遗忘带来的风险，进一步提升模型的泛化能力。

1 相关工作

本文提出的方法主要与基于度量学习的方法和基于提示学习的方法密切相关。

1.1 基于度量学习的方法

Koch 等[5]提出由两个结构相同且部分共享权重的网络构成的孪生网络模型，通过欧式方法计算输入样本对的匹配程度来判断它们是否属于同一类别。Vinyals 等[6]提出一种匹配网络模型，通过记忆网络和注意力机制，实现对以往知识的记忆存储，并快速学习新样本的特征。Snell 等[7]提出原型网络模型，将不同类别的平均向量作为类别原型的向量表示，最后在推理阶段，通过计算样本到类别原型向量的距离，实现对类别的预测。Sung 等[8]提出关系网络模型，该方法通过一个神经网络关系模块，实现自动学习特征间的距离度量关系表示。Geng 等[9]提出归纳网络模型，在关系网络的基础上，引入动态路由机制，实现获取支撑实例的类别向量表示，并通过关系模块计算查询实例与支撑实例的关系得分来进行分类。随后，Geng 等[10]又提出动态记忆归纳网络，通过引入二阶段训练范式，在第一阶段进行有监督的训练，为第二阶段的训练提供一个良好初始化的编码器和记忆模块，同时利用动态路由机制，为少样本学习提供更强大的灵活性，让模型更好地适应训练数据。

基于度量学习的方法采用传统度量或深度度量等方法，可以实现对类别的表征进行表示。但是，在不同的任务中，不同的度量方法差异性较大，这类方法无法适应多样化的实际问题。此外，基于度量学习的方法过于依赖训练数据，当数据较少时，不能很好地学习到类别的映射关系。

1.2 基于提示学习的方法

Schick 等[11−13]提出模式探索训练(pattern exploiting training, PET)方法，用于少样本学习。该方法通过定义并添加人工构建的模板，将文本分类任务转换为完形填空任务。在训练过程中，PET 方法将分类标签转换成标签描述形式，并使用[MASK]进行替换，填入人工定义的模板当中。通过语言模型还原[MASK]位置的词，最后使用标签映射策略完成文本分类任务。随后，Liu 等[14]在 PET 的基础上提出自适应 PET 模型，将模板中需要模型预测的词从有限候选词变成整个词表，通过扩大其搜索空间来增加模型的泛化性能，并且通过正确标签反向预测原文中的字符，进一步提升模型的性能。Gao 等[15]提出表现更好的少样本微调语言模型(LM-BFF)。该模型首先通过 T5 模型[16]实现自动化的生成最优模板，避免人工搜索模板这一繁杂的过程。接下来，将提示示例通过上下文的形式添加到原始输入中，利用更丰富的文本信息完成语言模型的建模工作。Liu 等[17]提出提示微调(P-tuning)模型。该模型丢弃提示模板必须是自然语言的假说，让语言模型自动学习适合当前任务形式的最佳提示模板形式。在训练过程中，使用预训练模型词表中未使用的字符去学习模板的连续表示形式，并且只学习更新模板对应的参数，从而极大地减小模型需要学习的参数量。

最近，Wang 等[18]提出少样本学习的蕴含 EFL 模型。与将文本分类任务转换为完形填空任务形式不同的是，EFL 是将文本分类任务转换为文本蕴含

任务形式。在训练过程中，对于每一个原始输入，EFL根据正确的标签描述生成新的正例，并根据其余候选标签随机生成若干新的负例。通过上述数据构造方式，实现原始输入与正确的标签描述模板构成蕴含关系，与其余候选标签则构成非蕴含关系。Jiang等[19]提出两种不同的模板集成方法：一种是概率平均的集成方法，通过训练集选择若干性能最好的提示模板，然后在推理阶段，将候选的若干提示模板的概率平均值作为最终预测结果；另一种是优化提示权重的集成方法，对于每一种关系引入可学习权重，最终输出概率为前若干提示输出概率的加权和。Hu等[20]提出一种知识型的提示学习调优方法，使用外部知识库扩展标签词空间，提高标签词的覆盖率，在零样本和少样本文本分类任务中证明了有知识调优的有效性。Min等[21]提出一种用于语言模型提示的噪声通道方法，证明使用计算给定标签输入通道的噪声通道的方法显著优于直接计算标签概率的方法。Zhang等[22]提出同时使用基学习器和元学习器的提示学习方法，证明度量学习可以帮助提示学习的方法更快地收敛。基于提示学习的方法是在大规模无监督语料训练的预训练语言模型基础上发展起来的，旨在减小预训练任务和下游任务形式之间的巨大差异，使下游任务形式尽可能与预训练任务形式保持一致。

2 EPL4FTC 算法

本文提出的EPL4FTC模型由基于自然语言推理的提示学习模块和度量优化模块两部分组成，两个模块共享编码层的参数。其中，基于自然语言推理的提示学习模块通过掩码语言模型头层计算输入句子中[MASK]位置处推理词的概率，并通过单句级和句群级两种粒度损失方法进行模型优化。度量优化模块对训练样本进行随机抽样，通过共享编码层编码后，使用三元组损失计算锚点与正负例之间的损失，最后对两个模块联合学习。

2.1 基于自然语言推理的提示学习模块

如图1所示，基于自然语言推理的提示学习模块负责将文本分类任务转换为基于自然语言推理形式的完型填空任务。具体做法是，对于原始输入文本，我们通过模板映射，将真实标签转化为自然语言推理形式。其中，推理词使用预训练语言模型中[MASK]字符替代，通过建模上下文间的关系，推理出[MASK]位置上真实的推理词。下面给出基于自然语言推理的提示学习方法的形式化表达。

对于给定的输入文本x，对应的真实标签为l，需要推理判断的标签描述为d。通过函数f_{prompt}，将输入x转换为基于提示学习的新的输入形式x'：

$$x' = f_{\text{prompt}}(x, z, d), \tag{1}$$

其中，z表示通过verbalize映射，将真实标签与需要推理判断的标签描述的关系转为逻辑推理词，可表示为

$$z = \text{verbalize}(l, d)。 \tag{2}$$

定义模板的一般形式为$[x']$ = “$[x]$, $[z]$$[d]$”。对于原始输入$x$，将其填充到$[x]$中，将需要推理判断的标签描述$d$填充到$[d]$中。接下来，通过映射函数verbalize($l$, d)，将输入x的真实标签描述l与当前填充需要推理判断的标签描述d转换为当前输入的逻辑推理词z。其中，$[z]$将被预训练语言模型中的[MASK]字符替代，逻辑推理关系词z将作为$[z]$的真实标签参与模型的优化。在推理阶段，通过映射函数f_{prompt}，将输入x和所有的标签描述d转化为基于提示学习的x'的形式。最后，通过计算$[z]$处的自然语言推理词概率，选取预测为蕴含关系最大概率的标签描述d对应的真实标签作为最终预测结果。

当采用自然语言形式的逻辑推理词时，使用自然语言中的“是”表示蕴含推理关系，“不是”表示非蕴含推理关系。进一步地，为了让语言模型学到更通用的自然语言推理表示，对推理词采用连续式的提示模板形式。也即，使用词表中未使用过的字符“[U1]”代表蕴含推理关系，“[U2]”代表非蕴含推理关系。

针对单样本输入形式以及通过数据增强形式扩增负样本形成的样例集合形式，设计两种粒度的损失函数来优化建模效果。

1) 单句级损失函数。如图2所示，对于每一个通过f_{prompt}映射函数构成的新的输入实例，需要模型完成建模上下文信息，预测推理出[MASK]位置处的真实推理词，并通过交叉熵进行优化。在给定输入x的情况下，定义[MASK]处推理词z的概率分布如下：

$$q(l|x) = \frac{\mathrm{e}^{s(z|x)}}{\sum_{z \in Z} \mathrm{e}^{s(z|x)}}, \tag{3}$$

式(3)中，Z表示候选推理词集合，$s(z|x)$ = MLM($z|f_{\text{prompt}}(x)$)表示在[MASK]处对候选推理词集合的

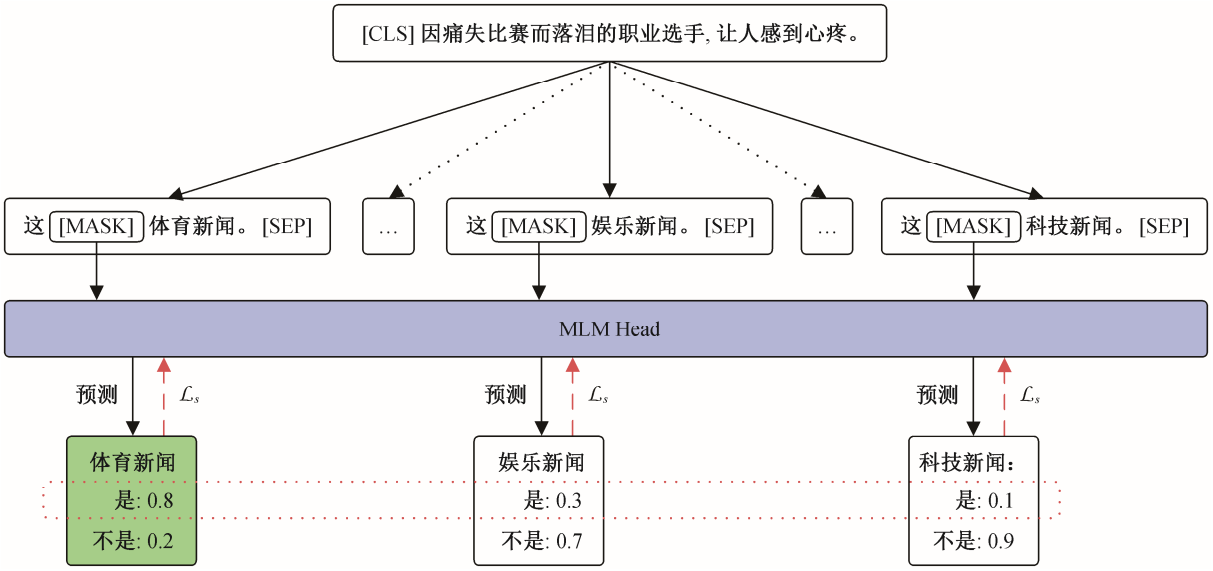

图 1 基于自然语言推理的提示学习结构图

Fig. 1 Structure of prompt learning based on natural language reasoning

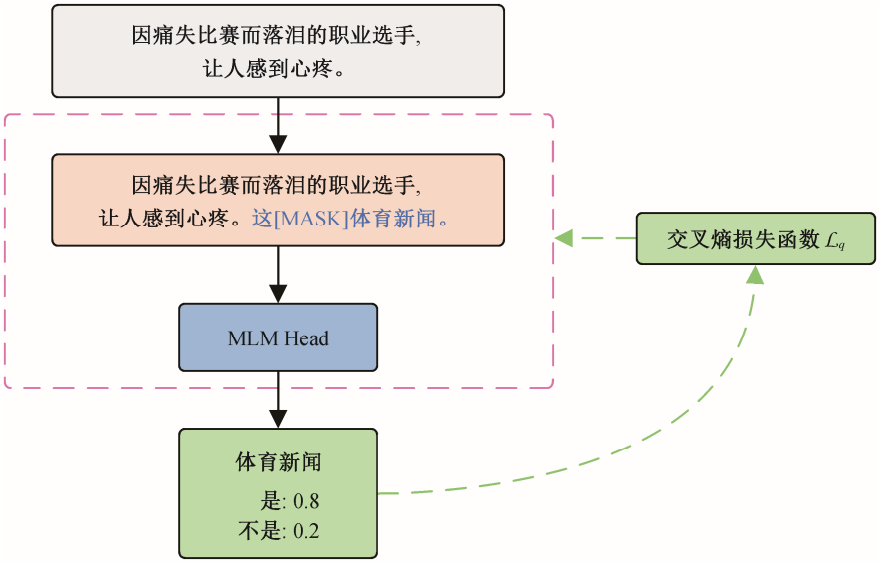

图 2 单句级的优化流程

Fig. 2 Sentence-individual level optimization

语言模型得分。最后，通过交叉熵损失计算单句级损失:

$$\mathcal{L}_s = \mathrm{CE}(q(l|x), z)。 \tag{4}$$

2) 句群级损失函数。单句级损失函数仅考虑对实例进行优化，没有考虑同一组正负样本间的关系，因此定义句群级损失函数，实现对一组正负样本间的关系进行优化，如图 3 所示。具体地，在对输入的实例进行数据构造时，通过输入实例与所对应的类别生成一个正例，将输入实例与其他类别进行数据构造，生成 $n-1$ 个负例，最终为每一条输入样本获得 n 个实例样本。最后，采用交叉熵损失对句群级进行优化:

$$\mathcal{L}_q = \mathrm{CE}(g(s(z|x)), I_{\mathrm{entail}}), \tag{5}$$

其中，I_{entail} 表示当前样例组中真实标签为蕴含关系的位置索引，$g(s(z|x))$表示语言模型对[MASK]处推理词在蕴含关系上的预测得分。最后，基于自然语

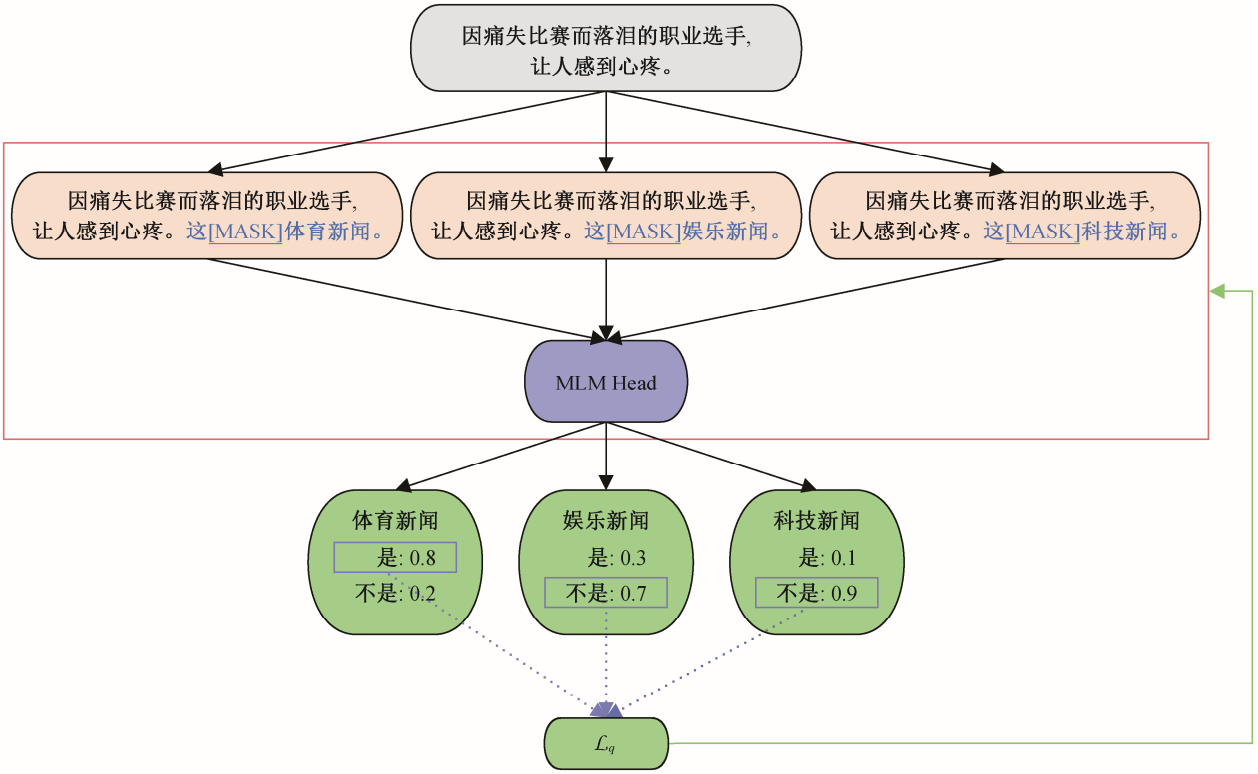

图 3　句群级的优化流程

Fig. 3　Sentence-group level optimization

言推理的提示学习模块的损失函数定义如下：

$$\mathcal{L}_{\mathrm{p}} = (1-\alpha)\mathcal{L}_{\mathrm{s}} + \alpha\mathcal{L}_{\mathrm{q}}, \tag{6}$$

其中，α为可调节的超参数。

2.2　度量优化模块

提示学习利用预训练语言模型在预训练任务中学习到的先验知识，在下游任务中可以取得一个良好的性能。但是，对文本分类任务而言，类别特征的表示也至关重要。通过度量学习，将原始语义空间的实例映射到目标任务中语义空间的表示，使实例在目标任务中的语义空间表示具有更强的区分能力。

度量优化模块的目标是使在语义空间中属于同一类别的实例之间的距离更接近，使不同类别实例之间的距离更远。通过三元组损失函数进行有监督的度量学习，使模型可以更好地学习不同类别间的距离关系信息[23]。此外，使用带间隔的损失函数，可以提升模型的泛化性能，如式(7)所示。具体地，在构造三元组数据时，在某个类别中选定一个实例作为锚点，同类别的实例作为正例，其他类别的实例作为负例。

$$\mathcal{L}_{\mathrm{tml}} = \sum_{m=1}^{M} \max(0, d(A_{\mathrm{m}}, P_{\mathrm{m}}) - d(A_{\mathrm{m}}, N_{\mathrm{m}}) + \Delta), \tag{7}$$

其中，$d(A_{\mathrm{m}}, P_{\mathrm{m}})$表示锚点与正例间的距离，$d(A_{\mathrm{m}}, N_{\mathrm{m}})$表示锚点与负例间的距离，$\Delta$表示设定的间隔值。

此外，在少样本学习场景中，用于训练的数据量通常十分有限。为了缓解灾难性遗忘的问题，使用掩码语言模型优化目标作为正则项进行建模。因此，度量优化模型的损失函数表示为

$$\mathcal{L}_{\mathrm{aux}} = (1-\beta)\,\mathcal{L}_{\mathrm{tml}} + \beta\mathcal{L}_{\mathrm{mlm}}, \tag{8}$$

式(8)中，$\mathcal{L}_{\mathrm{mlm}}$表示语言模型损失，$\beta$表示相应的权重参数。

最后，整体的损失函数由提示学习损失$\mathcal{L}_{\mathrm{p}}$和度量优化损失$\mathcal{L}_{\mathrm{aux}}$的加权构成：

$$\mathcal{L}_{\mathrm{total}} = (1-\gamma)\mathcal{L}_{\mathrm{p}} + \gamma\mathcal{L}_{\mathrm{aux}}, \tag{9}$$

其中，γ表示权重参数。

2.3　模型训练与推理

EPL4FTC算法将文本分类任务转化成自然语言推理任务，转化后的任务是一个二分类任务。因此，当一个原始分类任务包括N个类别时，该算法

需要进行N次推理, 最后选择预测概率最大的推理词对应的标签类别作为最终预测结果。为提升模型的泛化性能, 同时降低模型训练的成本, EPL4FTC算法通过负采样的方式对下游任务进行训练。对于一个包含多个类别的分类任务, 将每一个实例与之对应的类别作为正例, 同时随机选择K个其他类别与当前实例构成负例。上述数据构造方式不但能够提升模型的性能, 而且与使用全部类别作为负例相比, 进一步缩短了训练模型所需的时间。在模型推理阶段, EPL4FTC算法仅使用基于自然语言推理的提示学习模块。具体地, 对于包含N个标签的文本分类任务, 对每一个实例生成包含自然语言推理提示模板的N条新的输入实例。通过模型预测每一个实例中[MASK]处所蕴含推理词的概率, 在N个预测结果中选择预测概率最大的推理词对应的标签作为当前原始输入实例的预测结果。

3 实验与结果

3.1 实验数据集

1) 中文数据集。本文使用少样本评测数据集FewCLUE[24]中文本分类任务对应的数据集, 在4个领域的评测数据集上进行实验。其中, EPRSTMT为电商评论领域的情感分析任务, 是典型的包含正向和负向情感的二分类任务; CSLDCP是科学文献领域的长文本多分类任务, 包含67个类别; TNEWS是新闻标题的短文本分类任务, 包含教育、娱乐和文化等15个类别; IFLYTEK是根据APP应用的长文本主题描述信息, 对超过100多个应用类别进行分类的任务。

2) 英文数据集。本文采用3个英文文本分类数据集AG News, TREC和Yelp Review进行评测。其中, AG News[25]是学术新闻搜索引擎从多个新闻来源中搜集的超过100万篇文章构成的数据集, 包含世界、体育、商业和科技4类新闻主题; TREC[26]数据集包含6个一级标签和47个二级标签; Yelp Review数据集来自Yelp的用户评论, 其标签是用户对商品的星级打分, 共分为5级。用于评测的英文数据集从以上数据集中抽样获得。在3个原始英文数据集中分别随机抽取8个、16个和32个实例, 形成多个不同规模的数据集用于训练, 测试集为原始数据集中的测试集。

3.2 基线方法

1) 基于微调的方法(Fine-tuning)[3]: 在预训练语言模型的基础上, 为模型添加任务相关的分类器, 使模型可以处理下游任务。

2) Zero-shot方法[27]: 基于Roberta等自编码预训练语言模型, 通过MLM进行推理评测。

3) Zero-shot (GPT)方法[17]: 基于GPT自回归预训练语言模型[28], 通过从左至右的语言模型进行推理评测。

4) PET方法[12]: 通过添加人工自定义模板, 将下游任务转化成完成填空形式的任务, 然后在候选标签列表中选择合适的标签。

5) ADAPET方法[29]: 为模板搜索正确答案时, 从有限候选词变成整个词表, 扩大了模型的搜索空间。此外, 对正确标签反向预测原文中的词, 实现模型性能的提升。

6) LM-BFF方法[15]: 将自动化生成的离散化自然语言作为提示模板, 同时通过采样的形式, 将实例以上下文的方式添加到每一个输入中。

7) P-tuningR方法[17]: 有别于自然语言形式的提示模板, P-tuningR采用Roberta作为预训练语言模型, 使模型自动学习到最佳的连续式的非自然语言提示模板。

8) EFL方法[25]: 通过添加人工自定义模板, 将下游任务转化成蕴含任务形式, 并添加额外的二分类器, 实现对下游任务的微调。

3.3 实现细节与评测指标

实验在配有CUDA环境的Linux操作系统中进行, 并配置两块GTX 1080Ti显卡。代码使用基于PyTorch[30]框架的HuggingFace工具包来实现。对于中文数据集的评测, 采用12层网络结构的中文RoBERTa-wwm-ext[31]预训练模型; 对于英文数据集的评测, 采用12层结构的BERT-BASE预训练模型。模型参数设置如下: 学习率为10^{-5}, 超参数设置为$\alpha = 0.7$, $\beta = 0.01$, $\gamma = 0.02$, 三元损失间隔$\Delta = 0.15$, 并使用AdamW[32]优化器进行模型参数的优化。在少样本学习问题中, 通常使用准确率(Accuracy)作为评测指标, 表示模型预测正确的样本数量占所有样本数量的比例。

3.4 实验结果

1) 中文数据集的实验结果。如表1所示, 基于微调的方法在小样本学习场景中模型性能表现不佳。对于基于提示学习的方法, PET, LM-BFF, EFL, P-tuningR以及EPL4FTC算法在小样本学习场景中模型的准确率都大幅度提高, 表明基于提示学习的

表 1　中文少样本数据集实验结果

Table 1　Experimental results on Chinese few-shot datasets

方法	准确率/%				
	EPRSTMT	CSLDCP	TNEWS	IFLYTEK	平均值
人工	90.0	68.0	71.0	66.0	73.8
Fine-tuning	65.4	35.5	49.0	32.8	45.7
Zero-shot	85.2	12.6	25.3	27.7	37.7
Zero-shot (GPT)	57.5	26.2	37.0	19.0	35.0
PET	**85.6**	51.7	53.7	35.0	56.5
ADAPET	85.1	43.5	46.4	36.6	52.9
LM-BFF	84.6	54.4	53.0	46.1	59.5
P-tuningR	80.6	54.8	52.2	**48.0**	58.8
EFL	85.0	45.0	52.1	42.7	56.2
EPL4FTC (本文方法)	85.3	**55.1**	**54.6**	46.4	**60.4**

说明: 粗体数字表示性能最佳, 下同。

方法具有强大的潜能。对比EPL4FTC算法与其他基于提示学习的方法(PET, ADAPET, LM-BFF, EFL和P-tuning等)可以看出, EPL4FTC算法在EPRSTMT, CSLDCP和TNEWS等数据集上取得优异的成绩, 在IFLYTEK数据集上也取得与其他现有方法同等的性能。并且, EPL4FTC算法在中文文本分类任务中的平均准确率取得最高的成绩。与转换为完形填空任务形式的PET和ADAPET等方法相比, EPL4FTC算法在利用预训练模型中学习到的通用知识的基础上, 引入下游任务的类别信息, 实现更好的建模效果, 并且平均准确率高出3.9%。与转化为文本蕴含任务的EFL方法相比, EPL4FTC算法没有引入额外需要学习的大规模参数, 并且与预训练语言模型的任务保持一致, 有效地减小了上下游任务间的差异, 最终平均准确率高出4.2%。与使用自动构建模板或非自然语言形式模板的LM-BFF 和P-tuning方法相比, EPL4FTC算法无需繁琐的模板构建形式, 并且平均准确率高出1.6%。

2) 英文数据集的实验结果。如表2所示, 对于不同的实例数量, Fine-tuning, PET, ADAPET, EFL, P-tuning以及EPL4FTC算法都表现出随着实例数量增多, 模型准确率都明显提升的趋势, 表明在基于深度模型的少样本学习场景中, 训练数据的规模对模型性能有较大的影响。在实例数K = 8时, 虽然PET, ADAPET, EFL和P-tuning等基于提示学习的方法比基于微调的方法在准确率方面有很大幅度的提升, 但EPL4FTC算法表现出更加出众的性能, 其准确率远高于其他方法。这表明在实例较少的情况下, EPL4FTC算法能够有效地对下游任务进行建模, 也进一步证明了该算法的有效性。随着实例数增加(K = 16或32), 虽然其他基于提示学习方法的性能也有所提升, 但相比于其他方法, EPL4FTC算法的准确率仍然保持较高的水平。即使在K = 32的情况下, EPL4FTC算法的性能也与现有模型保持在同一水平, 并且平均准确率最佳。

3.5 组件有效性分析

3.5.1 度量优化模块有效性

在基于度量学习的损失优化方法对比实验中, 对比以下3种优化方法: 1) 将欧式距离和余弦相似度作为度量方法的二元交叉熵损失优化方法; 2) 对比损失优化方法; 3) 三元组损失优化方法。

在使用二元交叉熵损失作为损失优化的实验中, 采用欧式距离作为度量方法。由于其度量值域范围是[0, +∞), 为便于计算二元交叉熵损失, 将其映射到值域空间[0, 1)的范围:

$$f(r)=\tanh\left(\frac{1}{r+\epsilon}\right), \tag{10}$$

其中, r表示欧氏距离, 引入超参数ϵ是为了避免分母为0。

在使用基于余弦相似度的度量方法中, 其值域范围是[−1, +1]。同理, 将其映射到值域空间[0, 1]的范围:

$$f(r)=\frac{1}{2}(r+1)。 \tag{11}$$

其中, r表示余弦相似度。

表 2 英文少样本数据集实验结果

Table 2 Experimental results on English few-shot datasets

实例数	方法	准确率/%			
		AG News	TREC	Yelp Review	平均值
$K=8$	Fine-tuning	52.5	29.2	18.7	33.5
	PET	76.0	38.8	25.0	46.6
	ADAPET	78.8	21.6	24.2	41.5
	P-tuningR	68.3	31.8	22.2	40.8
	EFL	78.3	41.6	21.2	47.0
	EPL4FTC (本文方法)	**79.5**	**55.8**	**26.1**	**53.8**
$K=16$	Fine-tuning	66.4	46.2	26.5	46.4
	PET	83.3	62.4	30.3	58.7
	ADAPET	83.6	62.7	31.7	59.3
	P-tuningR	83.2	66.4	30.0	59.9
	EFL	84.6	55.4	30.0	56.7
	EPL4FTC (本文方法)	**84.9**	**68.6**	**32.3**	**61.9**
$K=32$	Fine-tuning	80.8	66.8	32.9	60.2
	PET	84.7	80.3	39.0	68.0
	ADAPET	85.8	80.1	31.5	65.8
	P-tuningR	86.1	**81.8**	39.2	68.4
	EFL	86.0	81.0	35.9	67.6
	EPL4FTC (本文方法)	**86.2**	80.8	**40.0**	**69.0**

实验结果如表 3 所示。可以看出，将欧式距离或余弦相似度作为度量方法的二元交叉熵损失优化方法性能较差，而对比损失优化方法和三元组损失优化方法的性能有较大的提升，得益于后两个方法中引入间隔的策略，使模型有了一定的容错空间，进而提升了模型的泛化性能。对比 3 组不同的实例可以看出，三元组损失优化方法可以同时获取更多的信息来帮助模型优化，从而提升模型性能。与对比损失优化方法相比，三元组损失优化方法在不同任务中的平均准确率有 1.4%的提升。

对采用三元组损失的度量优化模块进行消融实验，结果如表 4 所示。可以看出，将度量优化模块完整地移除后，模型的准确率明显下降，在中文数据集中平均下降 1.6%，在英文数据集中平均下降 3.2%，验证了度量优化模块的有效性。度量优化模块通过学习下游任务中的类别信息，实现对模型性能的提升。进一步，在度量优化模块中将 MLM 损失作为三元组损失的正则项引入。为了验证 MLM

表 3 中文数据集和英文数据集上不同损失优化实验结果

Table 3 Experimental results of different loss optimization methods on Chinese datasets and English datasets

方法	准确率/%								
	中文数据集					英文数据集			
	EPRSTMT	CSLDCP	TNEWS	IFLYTEK	平均值	AG News	TREC	Yelp Review	平均值
BCE Loss (CS)	77.6	54.3	54.1	45.2	57.8	72.2	53.4	25.3	50.3
BCE Loss (ED)	80.6	**55.5**	52.8	45.1	58.5	72.1	51.6	25.1	49.6
Contrastive Loss	83.2	54.5	53.9	44.4	59.0	72.9	54.0	**26.6**	51.2
Triplet Loss	**85.3**	55.1	**54.6**	**46.4**	**60.4**	**79.5**	**55.8**	26.1	**53.8**

说明: BCE Loss (CS)代表将余弦相似度作为度量的二元交叉熵损失优化方法, BCE Loss (ED)代表将欧式距离作为度量的二元交叉熵损失优化方法, Contrastive Loss 代表对比损失优化方法, Triplet Loss 代表三元组损失优化方法。

表 4　中文数据集和英文数据集上度量优化模块消融实验结果

Table 4　Ablation results of metric optimization module on Chinese datasets and English datasets

方法	准确率/%								
	中文数据集					英文数据集			
	EPRSTMT	CSLDCP	TNEWS	IFLYTEK	平均值	AG News	TREC	Yelp Review	平均值
EPL4FTC (− Triplet Loss − MLM)	83.4	54.6	54.0	43.3	58.8	75.5	54.4	21.7	50.6
EPL4FTC (− MLM)	85.2	53.0	54.2	45.7	59.5	78.8	50.0	26.3	51.7
EPL4FTC	85.3	55.1	54.6	46.4	60.4	79.5	55.8	26.1	53.8

正则项的有效性, 实验中仅保留三元组损失, 并移除 MLM 损失正则项。从实验结果可以看出, 移除 MLM 正则项后, 模型的准确率在大部分任务中都明显下降, 在中文 CSLDCP 任务中下降 2.1%, 在英文 TREC 任务中下降 5.8%, 证明了引入 MLM 损失作为正则项对模型性能提升的有效性。

3.5.2 句群级损失有效性

基于自然语言推理的提示学习模块中, 通过句群级损失实现对一组正负实例间的优化。为了确定该损失优化方法的有效性, 对其进行消融实验, 结果如表 5 所示。中文数据集的实验结果显示, 对于不同的任务, 该方法对模型的性能都有明显的提升, 特别是对于 IFLYTEK 任务, 模型的性能有 3%的提升。相比于中文数据集, 在英文数据集实例数 K=8 时的实验结果显示, 该损失优化方法对模型的准确率具有更大的提升作用, 在 AG News 数据集上显著提升 38.6%; 在 Yelp Review 数据集上也有 5.9%的提升。上述实验结果证明了句群级损失方法对组内优化的有效性, 它通过对比组内正负间的实例, 可以学习到更好的知识表示。

3.6 提示模板分析

3.6.1 推理词形式的性能

EPL4FTC 算法将文本分类任务转换为基于自然语言推理形式的完型填空任务, 同时受 P-tuning 方法启发, 推理词不仅可以是自然语言形式, 也可以是非自然语言形式。因此, 本文对这两种形式的推理词进行性能评估。中、英文数据集的实验结果(表 6)表明, 非自然语言形式的推理词较为稳定, 模型的性能较好。具体地, 对于形式简单、数据区分度高的任务(如 EPRSTMT 和 TREC 等), 自然语言形式的推理词表现较为出众; 对于类别数较多、复杂的任务(如 TNEWS, IFLYTEK 和 CSLDCP 等), 非自然语言形式的推理词具备更好的性能。这是由于它可以从具体任务中自主地学习到更适合当前模板的推理词形式, 而不受自然语言形式的限制。也就是说, 对于非自然语言形式的推理词, 可以从众多的

表 5　中文数据集和英文数据集上句群级损失有效性分析

Table 5　Validity analysis of sentence-group level loss on Chinese datasets and English datasets

方法	准确率/%						
	中文数据集				英文数据集		
	EPRSTMT	CSLDCP	TNEWS	IFLYTEK	AG News	TREC	Yelp Review
EPL4FTC	85.3	55.1	54.6	46.4	79.5	55.8	26.1
w/o L_{group}	84.9	54.2	54.1	43.4	40.9	39.6	20.2

表 6　中文数据集和英文数据集上推理词形式性能比较

Table 6　Performance comparison of inference word form on Chinese datasets and English datasets

方法	准确率/%								
	中文数据集					英文数据集			
	EPRSTMT	CSLDCP	TNEWS	IFLYTEK	平均值	AG News	TREC	Yelp Review	平均值
自然语言推理词	86.0	54.3	53.5	45.4	59.8	77.3	59.0	26.1	54.1
非自然语言推理词	85.3	55.1	54.6	46.4	60.4	79.5	55.8	26.1	53.8

上下文信息中学习到推理词的连续化表达形式，有效地避免单一推理词的影响。

3.6.2 提示模板的性能

手工设计的提示模板会使模型的效果产生一定的波动，因此本文评估手工设计模板对模型性能的影响，实验结果如表 7 所示。可以看出，模型性能受提示模板的影响较大。具体地，在中文 TNEWS 和英文 TREC 任务中对模板采用前缀式与后缀式的形式进行评测，相比之下，在中文数据集上模型的性能差异相对较小，准确率的最大值与最小值相差 1.1%，而在英文数据上模型的性能表现出较大的差异，准确率最大值与最小值相差 6.4%。上述实验结果表明，提示模板对模型准确率的影响与下游任务的具体形式有较大的关系，可以通过优化模板的形式来大幅度提升模型的性能。

3.7 可视化分析

为了评估引入度量优化模块后提出的模型获得任务类别信息的有效性，本文采用 t-SNE 方法[33]，对中文 TNEWS 数据集通过随机采样进行可视化分析。为了验证模型的编码层是否有效地学习到任务

表 7 准确率受提示模板的影响

Table 7 Accuracy with different prompt templates

TNEWS		TREC (K = 8)	
模板	准确率/%	模板	准确率/%
下面<MASK><desc>新闻: <text>。	54.2	This <MASK> the <desc> question: <text>.	50.0
<text>, <desc>新闻？<MASK>。	54.6	<text>, it <MASK> about <desc> question.	56.4
<text>, 这<MASK><desc>新闻。	53.9	<text>. it <MASK> <desc> question.	55.8

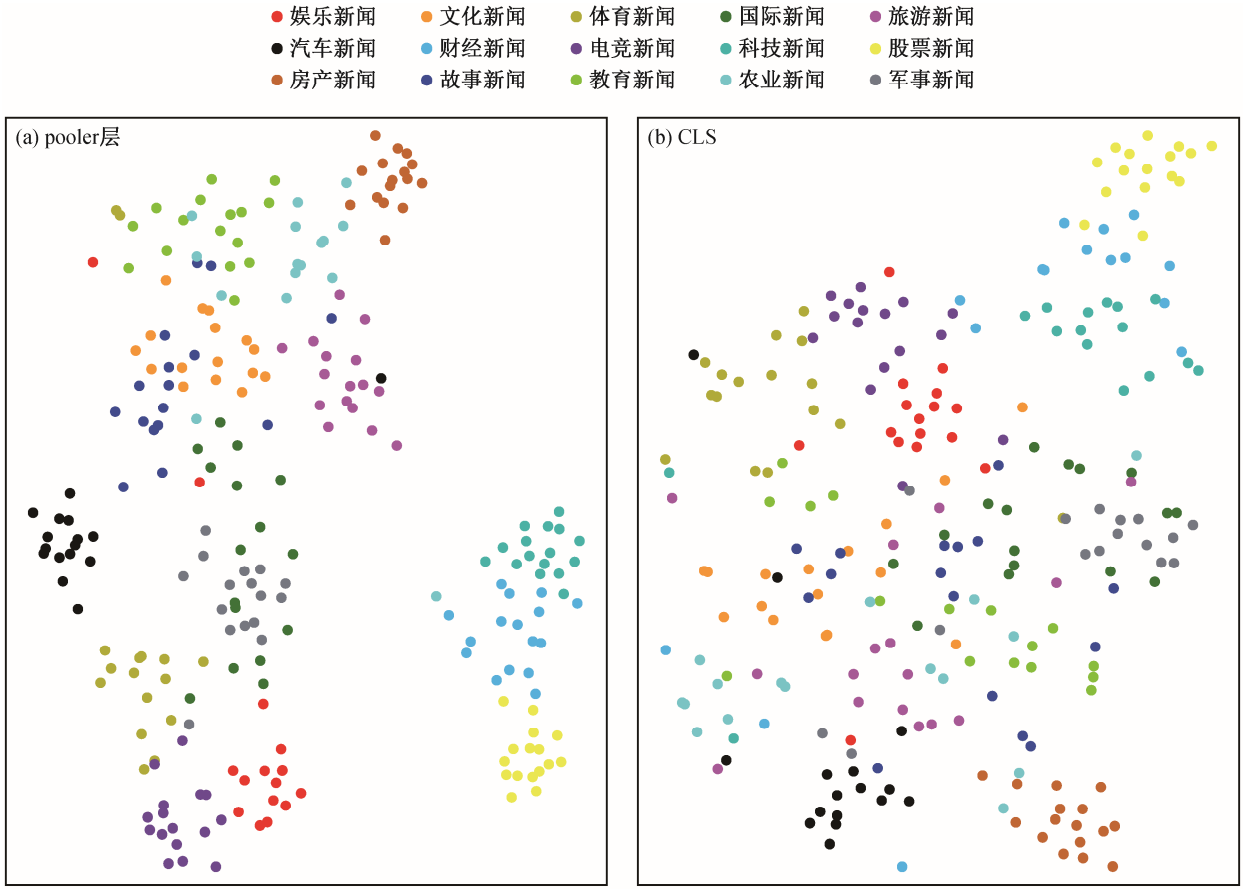

图 4 实例向量的 t-SNE 分布可视化

Fig. 4 Visualization on instance vectorization with t-SNE distribution

中的类别信息, 将预训练模型中编码层CLS位的输出作为当前整个实例的向量化表示。图4(a)展示编码后的分布情况, 可以看出实例类别依然保持与pooler层相似的分布情况。对于简单的新闻类别(如股票、娱乐、电竞和汽车等), 实例类别依旧保持着较为紧凑的聚集现象; 对于较为抽象或涵盖范围较广的新闻类别(如文化和故事等), 虽然实例类别的分布较为分散, 但也存在一定程度的区域性。这表明CLS作为整个句子的编码表示已经学习到一定的实例类别信息。对实例类别分布的可视化分析结果表明, 度量优化模块可以为模型提供更多额外的类别知识等信息。

在度量优化模块中, 采用三元组损失优化类别间的距离。具体地, 将预训练模型中pooler层的输出通过度量优化模块进行度量学习。图4(b)展示实例经该模块编码输出后的向量分布情况, 可以看出同一类别的实例间都较为紧凑, 同时不同类别的实例间也存在较为明显的间隔距离, 说明模型至少在pooler层中已经学习到非常好的类别表示。

4 结论

本文提出一种基于提示学习和三元组损失优化方法的少样本文本分类的EPL4FTC算法, 面对仅含有少量实例的文本分类任务, 该算法能够有效地完成文本分类。

本文利用提示学习, 将文本分类任务转换成自然语言推理形式, 通过提示学习激活预训练语言模型中已学习到的通用知识, 并通过句子和句群两种粒度的三元组损失优化方法, 实现捕获下游文本分类任务的类别表征, 提升文本分类的准确性。同时, 引入掩码语言模型任务的训练目标为正则项, 提升模型的泛化性能。

本研究完成在中、英文多个数据集上的实验, 结果表明文本分类的准确率有所提升, 验证了EPL4FTC算法的有效性。

未来的工作中, 我们将尝试将EPL4FTC算法应用于其他主题的少样本任务场景。此外, 对中英文之外其他语种的少样本文本分类研究也是一个有趣的问题。

参考文献

[1] Minaee S, Kalchbrenner N, Cambria E, et al. Deep learning-based text classification: a comprehensive review. ACM Computing Surveys, 2021, 54(3): 1–40

[2] Wang Y, Yao Q, Kwok J T, et al. Generalizing from a few examples: a survey on few-shot learning. ACM Computing Surveys, 2020, 53(3): 1–34

[3] Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding // NAACL. Minneapolis, 2019: 4171–4186

[4] Schroff F, Kalenichenko D, Philbin J. Facenet: a unified embedding for face recognition and clustering // CVPR. Boston, 2015: 815–823

[5] Koch G, Zemel R, Salakhutdinov R. Siamese neural networks for one-shot image recognition [D]. Toronto: University of Toronto, 2015

[6] Vinyals O, Blundell C, Lillicrap T, et al. Matching networks for one shot learning // NIPS. Barcelona, 2016: 3637–3645

[7] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning // NIPS. Long Beach, 2017: 4080–4090

[8] Sung F, Yang Y, Zhang L, et al. Learning to compare: relation network for few-shot learning // CVPR. Montreal, 2018: 1199–1208

[9] Geng R, Li B, Li Y, et al. Induction networks for few-shot text classification // EMNLP-IJCNLP. Punta Cana, 2020: 3904–3913

[10] Geng R, Li B, Li Y, et al. Dynamic memory induction networks for few-shot text classification // ACL. Seattle, 2020: 1087–1094

[11] Schick T, Schütze H. Few-shot text generation with pattern-exploiting training // EMNLP. Abu Dhabi, 2022: 390–402

[12] Schick T, Schütze H. Exploiting cloze-questions for few-shot text classification and natural language inference // EACL. Kyiv, 2021: 255–269

[13] Schick T, Schütze H. It's not just size that matters: small language models are also few-shot learners // NAACL. Mexico City, 2021: 2339–2352

[14] Liu H, Tam D, Muqeeth M, et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning // NeuIPS. New Orleans, 2022: 1950–1965

[15] Gao T, Fisch A, Chen D. Making pre-trained language models better few-shot learners // ACL. Bangkok, 2021: 3816–3830

[16] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research,

2020, 21: 1–67

[17] Liu X, Zheng Y, Du Z, et al. GPT understands, too [EB/OL]. (2023–03–18) [2023–06–24]. https://doi.org/10.48550/arXiv.2103.10385

[18] Wang S, Fang H, Khabsa M, et al. Entailment as few-shot learner [EB/OL]. (2021–04–29) [2023–06–18]. https://doi.org/10.48550/arXiv.2103.10385

[19] Jiang Z, Xu F F, Araki J, et al. How can we know what language models know?. Transactions of the Association for Computational Linguistics, 2020, 8: 423–438

[20] Hu Shengding, Ding Ning, Wang Huadong, et al. knowledgeable prompt-tuning: incorporating knowledge into prompt verbalizer for text classification // ACL. Dublin, 2022: 2225–2240

[21] Min S, Lewis M, Hajishirzi H, et al. Noisy channel language model prompting for few-shot text classification // ACL. Dublin, 2022: 5316–5330

[22] Zhang H, Zhang X, Huang H, et al. Prompt-based meta-learning for few-shot text classification // EMNLP. Abu Dhabi, 2022: 1342–1357

[23] Weinberger K Q, Saul L K. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research, 2009, 10(1): 207–244

[24] Xu L, Lu X, Yuan C, et al. Fewclue: a Chinese few-shot learning evaluation benchmark [EB/OL]. (2021–09–29) [2023–05–16]. https://doi.org/10.48550/arXiv.2107.07498

[25] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification // NIPS. Montreal, 2015: 649–657

[26] Li X, Roth D. Learning question classifiers // COLING. Taipei, 2002: 556–562

[27] Liu Y, Ott M, Goyal N, et al. Roberta: a robustly optimized Bert pretraining approach [EB/OL]. (2019–07–26) [2023–04–19]. https://doi.org/10.48550/arXiv.1907.11692

[28] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pretraining [EB/OL]. (2018–06–11) [2023–03–21]. https://openai.com/research/language-unsupervised

[29] Tam D, Menon R R, Bansal M, et al. Improving and simplifying pattern exploiting training // EMNLP. Punta Cana, 2021: 4980–4991

[30] Paszke A, Gross S, Massa F, et al. Pytorch: an imperative style, high-performance deep learning library // NeuIPS. Seattle, 2019: 8026–8037

[31] Cui Y, Che W, Liu T, et al. Pre-training with whole word masking for Chinese Bert. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 29: 3504–3514

[32] Loshchilov I, Hutter F. Decoupled weight decay regularization [C/OL] // ICLR. (2019–05–06) [2023–03–26]. https://openreview.net/forum?id=Bkg6RiCqY7

[33] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, 9(11): 2579–2605