文章编号:1007-5321(2019)06-0076-08 **DOI**:10.13190/j.jbupt.2019-149

用于文本分类的多探测任务语言模型微调

傅群超, 王 枞

(1.北京邮电大学 软件学院, 北京 100876; 2.北京邮电大学 可信分布式计算与服务教育部重点实验室, 北京 100876)

摘要: 预训练语言模型被广泛运用在多项自然语言处理任务中,但是对于不同的任务没有精细的微调. 针对文本分类任务,提出基于探测任务的语言模型微调方法,利用探测任务训练模型特定的语言学知识,可提高模型在文本分类任务上的性能. 设计了 6 个探测任务,覆盖句子浅层、语法和语义三方面信息. 最后在 6 个文本分类数据集上验证了本文的方法,使分类错误率得到改善.

关 键 词: 探测任务; 语言模型; 多任务学习; 文本分类

中图分类号: TN929.53 **文献标志码**: A

Based on Multiple Probing Tasks Fine-Tuning of Language Models for Text Classification

FU Qun-chao, WANG Cong

(1. School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. Key Laboratory of Trustworthy Distributed Computing and Service (Beijing University of Posts and Telecommunications), Ministry of Education, Beijing 100876, China)

Abstract: Pre-trained language models are widely used in many natural language processing tasks, but there is no fine-tuning for different tasks. Therefore, for text classification task, the author proposes a method of fine-tuning language model based on probing task, which utilizes the specific linguistic knowledge of probing task training model, and improves the performance of the model in text classification task. Six probing tasks are given to cover the shallow information of sentences, grammar and semantics. The method is shown validated on six text classification datasets, and classification error rate is improved.

Key words: probing task; language model; multiple task; text classification

从原始文本中学习有效信息的能力是减轻自然语言处理(NLP, natural language processing)对监督学习依赖的关键. 大多数深度学习方法需要大量手动标记的数据,这限制了它们在许多缺乏标准数据集上的适用性. 在这些情况下,如果可以利用未标记的语料学习语言信息来替代人工标注数据集是十分有吸引力的,因为人工标注数据集既耗时又昂贵. 此外,即使在可获得足够大的监督情况下,通过无监督的语言模型学习到良好的表示也可以显着的提升性能. 语言模型预训练已经证明可以改善许多自然语言处理任务[1-5].

然而,利用未标记语料学习语言信息有 2 个挑战. 首先,尚不清楚哪种类型的优化目标在传输有用的文本表示方面最有效;其次,对于使用哪种方法将这些学习的表示转移到目标任务是最有效的没有达成共识. 现有 2 种策略可用于将预训练语言表示

收稿日期: 2019-11-22
基金项目: 国家重点研发计划项目(2017YFC1307705)
作者简介: 傅群超(1992—), 男, 博士生, E-mail: fuqunchao@bupt.edu.cn; 王 枞(1958—), 女, 教授, 博士生导师.

应用于下游任务：基于特征的方法和基于微调(Fine-tuning)的方法，基于特征的方法将预训练语言模型学习到的表示作为特征用于下游任务，如ELMo(embeddings from language models)[4]，使用特定于任务的体系结构，将预先训练的文本表示作为附加特征，提高模型的性能；基于微调的方法，例如GPT(generative pre-trained transformer)[5]、BERT(bi-directional encoder representations transformers)[6]，此类方法引入了少量的任务特定参数，并通过简单微调参数使得预训练模型更适用于下游特定任务. 在之前的相关研究中，这 2 种方法都在预训练期间采用相似的目标函数语言模型来训练参数. 其中，GPT 使用单向语言模型来学习通用语言表示，而BERT 采用双向的语言模型.

大多数相关研究都集中在提高语言模型的文本表示能力，但是对于哪种表示特征有力于特定的下游任务处于一个较浅的层面. 对于不同的下游任务，预训练语言模型需要不同的优化目标，所以针对文本分类任务，提出新的基于探测任务的语言模型微调方案，利用探测任务优化语言模型，使其更适用于文本分类任务，最终提高文本分类任务的准确率. 文本分类是一项重要的自然语言处理任务，有多种实际应用，如垃圾邮件过滤、欺诈检测、机器人检测、法律发现等.

提出了一种新的半监督文本分类模型 PFTLM(probing task fine-tune language model)，结合无监督预训练语言模型和特定任务的有监督微调. 笔者的研究目标是学习一种适用于文本分类的表示学习和有监督微调方法. 假设有大量未标记文本和少量人工标注的数据集，同时不要求这些标注的数据集与未标记的语料库处于同一个领域. PFTLM 模型分为 3 个阶段：首先，在未标记的通用领域语料上训练语言模型，目的是学习到通用的文本表示；随后，使用探测任务微调预训练的语言模型，目的是学习特定的语言学知识；最后，在预训练的语言模型上使用有标注的数据集微调目标任务分类器.

采用双向 Transformer 网络[7]构建语言模型，它已被证明在各种任务上的表现都十分出色，如机器翻译[7]、文档生成[8]和语法分析[9]等. 与循环网络等方案相比，Transformer 能在提供并行计算能力的同时，捕捉文本中的长期依赖关系，从而在各种任务中实现强大的传输性能.

最后在 6 个被广泛使用的文本分类数据集上验证本文的模型的性能. 实验结果表明，PFTLM 模型优于高度工程化的模型，并且能在少量有标注的数据集上获得较好的效果.

1　相关工作

1) 自然语言处理半监督学习. 半监督学习的自然语言处理方法已经有很多相关研究人员进行了研究，并被广泛运用在序列标注[10]或文本分类[11]等任务中. 最初使用未标记的数据来计算单词级或短语级统计信息，然后将其作为有监督模型中的初始特征[12]. 在最近几年，大量的研究证明，使用从未标注语料中学习到的词嵌入能够提高各种任务的性能[13-14]. 然而，这种方法只能迁移单词级别的信息，不能获取到更高级别的语义信息. 近期的研究结果表明，已经能从未标注的语料中学习到短语级别或者句子级别的嵌入，并在各种任务中被用于编码文本[15].

2) 无监督预训练. 无监督预训练是半监督学习的一个特例，目标是为了找到一个更好的初始化参数，而不是修改监督学习任务. 研究结果[16]表明，预训练作为正则化方案，可以在深度神经网络中获得更好的泛化性能. 在最近的研究中，此类方法已被广泛运用于多种深度神经网络任务，如图像分类、语音识别、实体消歧和机器翻译. 与笔者研究相似的是使用语言模型对神经网络进行预训练，然后对有监督的目标任务进行微调. 这种方法的优势是预训练能学习到通用的语言学信息，对于特定的任务只需要学习少量的参数，减少对标注数据集的依赖. Dai 和 Howard 等[1-2]将这种方法用于改进文本分类. 然而，尽管预训练阶段有助于捕获一些语言信息，但是他们所采用的 LSTM(long short-term memory)网络[17]在捕获长距离信息的能力有限，且训练参数众多，训练效率低下. 相比之下，GPT[5]和BERT 模型[6]使用的 Transformer 网络[7]能够抓取到更长距离的语言信息，并在多个任务上取得最优的结果.

3) 微调. 微调已经被成功运用在相似任务间的迁移上，例如，可用在问答系统[18]、远程监督的情感分析[19]，还有机器翻译领域[20]. 但是该方法被证明在不相关的任务间无效[21]. Dai 等[1]的研究表明，一般的微调模型需要大量的领域内语料和大量标注的训练集，以达到好的效果，并且会有过拟合的问题. Howard 等[2]使用通用领域语料的预处理和

新颖的微调技术防止了过拟合,并且仅需要少量的样本也能获得最优的结果.

4) 探测任务. 探测任务常被用来评判句子向量是否抓取了对应的语言学知识[22-23],探测任务通过句子向量在相应数据集的准确率上判断该句子向量是否抓取到相应的语言学知识. 笔者的研究聚焦在哪种语言学知识对文本分类任务是有帮助的. 通过探测任务微调预训练的语言模型,使其学习到相应的语言学知识,最后通过实验验证该语言学知识是否对下游任务有帮助.

2 模型

给定一个静态的源任务 τ_S 和任意一个其他任务 τ_T,这里 $\tau_S \neq \tau_T$,目的是通过源任务 τ_S 提高目标任务 τ_T 的性能. 无监督的语言模型是源任务 τ_S 的一个理想选择. 语言模型能抓取到许多与下游任务相关的语言学特征,如长程依赖[24]、层次关系[25]. 进一步说,语言模型已经是机器学习对话系统等任务中的关键部分.

PFTLM 模型分为 3 个步骤,第 1 步是从大量无标注的语料中学习高可用性的语言模型;第 2 步是利用多探测任务微调语言模型,强化特定语言学特征;第 3 步是在下游任务上用微调的方法,针对有监督的文本分类任务调整模型. 这是通用的方法,能运用在各种大小的文档、不同的标签数目和类型. 使用同一个结构和训练过程,并且不需要额外的特征工程和额外的相关领域补充语料.

2.1 无监督预训练语言模型

对于无监督的语言模型,BERT[6] 采用了 MLM (masked language model) 作为预训练模型,取得了很好的效果. 但是模型收敛慢,计算代价高,所以采用了双向的 Transformer 网络训练语言模型. 假设给定一个无监督语料集 U,共有 n 个词 $\{t_1, t_2, \cdots, t_n\}$. 标准的前向语言模型如

$$p(t_1, t_2, \cdots, t_n) = \prod_{i=1}^{N} p(t_i \mid t_{i-k}, \cdots, t_{i-1}) \quad (1)$$

其中 k 表示上下文窗口的大小.

反向语言模型与前向模型相似,不同之处在于从后向前遍历语料,如

$$p(t_1, t_2, \cdots, t_n) = \prod_{i=1}^{N} p(t_i \mid t_{i+1}, \cdots, t_k) \quad (2)$$

结合前向、后向模型,目标函数为如下对数似然函数:

$$L(U) = \sum_{i=1}^{n} (\log p(t_i \mid t_{i-k}, \cdots, t_{i-1}; \overrightarrow{\theta}, \theta_x, \theta_s) + \log p(t_i \mid t_{t+1}, \cdots, t_k; \overleftarrow{\theta}, \theta_x, \theta_s)) \quad (3)$$

其中 k 表示上下文窗口的大小;$\overrightarrow{\theta}$、$\overleftarrow{\theta}$ 分别表示前向和后向 2 个网络层的参数,并共享词表示 θ_x 和 Softmax 层参数 θ_s. 模型训练方式采用 Adam[26] 优化算法.

在实验中,使用多层 Transfromer 解码器[8] 建模语言模型,并使用 BooksCorpus[27] 数据集训练语言模型. 该数据集包含超过 7 000 本小说,涉及许多不同的类型. 而文本分类数据集来自于其他领域,下文会详细介绍所使用的数据集. 虽然这一步生成语言模型十分耗时,但只需要执行 1 次.

2.2 基于多探测任务的语言模型微调

预训练语言模型能够抓取到通用的语言学特征,但是不同类型的自然语言处理任务需要用到不同的语言学知识. 对于何种语言学知识有利于何种自然语言处理任务尚不明确. 通过探测任务使得语言模型学习到指定的语言学知识,并分析这些语言学知识是否对下游任务有所帮助. 模型采用多任务学习模式,同时优化多个探测任务的目标函数,使语言模型学习到多个语言学知识,提高模型泛化能力.

探测任务从序列标注,句子分割到句子对关系判断有很多种. 在构建探测任务过程中基于如下原则:首先需要能够快速构建大量的训练集,因为语言模型参数众多,需要足够的数据集来微调参数;其次是选择特点鲜明的语言学特征,有助于之后的消融分析.

首先,对于句子浅层的知识,构建了句子长度任务和单词蕴含任务. 句子长度任务是预测句子包含多少个单词,将句子按照长度分成 6 等份,所以该任务相当于一个 6 分类任务. 单词蕴含任务是预测一个句子是否包含一个单词,选取了 800 个中频词汇,并抽取了同等数量的包含这些词和不包含这些词的句子,每个样本就是一个二分类任务.

其次,对于语法层面的语言学知识,包括语序探测任务,调换其中一半样本的某 2 个单词的顺序,预测样本是否被打乱过,该任务目的是训练模型对语序的敏感程度;另一个是句子深度任务,目的是学习推断句子结构的能力,抽取了句子深度在 5 ~ 12 之间的句子,任务就是将一个句子分到这 8 类中.

最后,对于语义层面的语言学知识,有主语识别任务,该任务重点是判断主从句中有多少个主语.

为了防止模型依赖于特定的单词，训练集和验证集中没有重复的目标. 该任务是标注任务，标注句子中每个单词是否是主语. 相似地，有宾语识别任务，目的是使训练集和验证集避免出现重复的目标.

综上，设定了共 6 个探测任务，分别让语言模型学习到句子长度、单词蕴含、语序信息、句子深度识别、主谓语识别等语言学知识，并作用于下游文本分类任务.

这一阶段采用多任务训练，多个探测任务共享语言模型参数. 受 Howard 等[2]工作的启发，在模型优化阶段改进了分层微调和斜三角学习速率，下面介绍这 2 个改进后的方法.

1）分层微调

Yosinski 等[28]的研究表明，神经网络中不同的层抓取不同的信息，所以神经网络中不同层使用相同的学习速率是不合适的. 分层微调能够在不同的层上使用不同的学习速率. 假设模型共有 L 层，模型的参数 θ 可分为 $\{\theta^1,\theta^2,\cdots,\theta^L\}$，$\theta^l$ 表示模型第 l 层的参数. 相似地，学习速率分为 $\{\eta^1,\eta^2,\cdots,\eta^L\}$. η^l 表示模型第 l 层的学习速率. 分层微调在时刻 t 更新模型参数形式如下：

$$\theta_t^l=\theta_{t-1}^l-\eta^l\ \nabla_{\theta^l}J(\theta) \tag{4}$$

其中 $\nabla_\theta J(\theta)$ 为目标函数对于参数 θ 的导数.

语言模型包含多层 Transfromer 结构，每一层都学习不同的信息，如果采用相同的学习速率，会导致部分层参数没有优化到最低点或者脱离最低点，通过分层微调可以使语言模型的每一层都能优化到各自较优的区域.

2）斜三角学习速率

为了快速使语言模型获取探测任务相关的特征，使得模型能够快速收敛到合适区域，之后再精细调整参数. 采用斜三角学习速率，前期快速增长，后期慢慢衰减. 学习速率变化如图 1 所示.

太小的学习速率会让模型的优化过于谨慎，减慢训练速度. 而太大的学习速率会导致损失函数震荡，甚至难以收敛. 预训练的语言模型微调过程不需要太长的训练时间，所以需要提高学习速率让模型尽快进入参数较好的范围，然后降低学习速率，使得模型不会在最小值附近大幅震荡. 经过实验，最大学习速率 $\eta_{\max}$ 设置为 0.01，优化过程中前 10% 的时间学习速率逐渐递增至最大学习速率，后 90% 的时间逐步递减.

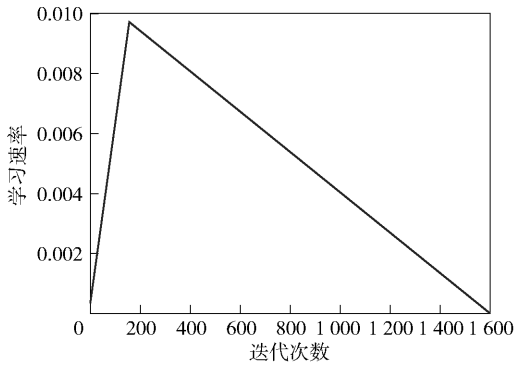

图 1　斜三角学习速率

2.3　有监督的分类器微调

模型要为目标任务调整参数. 假定有一个有标注的数据集 C，C 中每个实例都包含一个输入序列 $\{x_1,x_2,\cdots,x_m\}$ 和一个标签 y. 输入序列经过预处理的语言模型从最终层获取到输出 h_m，然后输入到一个线性输出层来预测 y.

$$p(y\mid x_1,x_2,\cdots,x_m)=\text{softmax}\ (h_mW_y) \tag{5}$$

其中 W_y 表示线性输出层的参数.

文本分类任务中的关键信息通常包含在几个单词中，这些单词可能出现在文档中的任何位置. 由于输入文档可能包含数百个单词，如果只考虑模型的最后隐藏状态，有效信息可能会丢失. 出于这个原因，将文档的最后一个步骤的隐藏状态 h_m 与各个步骤隐藏状态的平均 h_{avg} 相结合. 所以最后线性模型的输入是：$h_c=[h_m,h_{\text{avg}}]$，[ ] 表示链接操作. 式(5)修改为式(6).

$$p(y\mid x_1,x_2,\cdots,x_m)=\text{softmax}\ (h_cW_y) \tag{6}$$

微调目标分类器是模型最关键的一步. 过度激进的微调将导致灾难性的结果，失去语言模型抓取有效信息的优势；过于谨慎的微调会导致收敛缓慢以及过度拟合. 这一步仍然使用分层微调和斜三角学习速率. 除此之外还采用逐级解冻法.

逐级解冻法从最后一层开始逐级解冻模型，而不是一次性微调所有层，因为一次性微调所有层会导致灾难性的遗忘，逐级解冻至少能保留通用的知识. 首先解冻最后一层，在一个训练周期对所有未冻结层进行微调；然后解冻下一个较低的冻结层，并重复操作，直到对所有层进行微调. 值得注意的是该方法是一次向“解冻”层添加一个层，而不是一次只训练一个层.

3 实验及结果

虽然微调方法同样适用于其他自然语言处理任务,但笔者专注于文本分类任务. 文本分类任务在显示世界中有很多重要的应用,如情感分析、问题分类和主题分类等.

3.1 预训练语言模型

使用 BooksCorpus 数据集训练语言模型. 该数据集包含超过 7 000 本小说,涉及许多不同的小说类别,如冒险、奇幻和浪漫类. 重要的是该数据集包含足够长度的文本,可提供学习长距离信息的条件.

3.2 多探测任务

探测任务数据集中的所有句子来源于 BooksCorpus 数据集,并抽样选取了句子长度在 5 ~ 30 个词之间的句子,对于每个探测任务,构建了 10 万条数据量的训练集和 1 万条数量的验证集,并且每个训练集标注项都是平衡的.

3.3 文本分类数据集

在 6 个被广泛研究的数据集上评估所提出的模型,这些数据具有不同的文档数量和文本长度,可分为 3 个常见的文本分类任务,分别为情感分析、问题分类和主题分类.

情感分析数据集包括 IMDb (internet movie database)数据集[29]和 2 个不同版本的 Yelp 评论数据集[30]. IMDb 数据集中训练和测试数据各有 2.5 万条,分为正面或者负面. 使用了 2 个版本的 Yelp 数据集,分别是一个二分类数据集,样本数量为 56 万条;一个 5 分类数据集,样本数量为 65 万条.

问题分类数据采用 TREC (text retrieval conference)数据集[31],TREC 数据集是一个小型开发领域的数据集. 该数据集有 6 个分类,包括人、位置、数字信息等,共有 5 500 条训练样本. 该任务就是把基于事实的问题归到 6 个语义分类中.

主题分类数据集使用 AG(Antonio Gulli)新闻数据集合 DBpedia Ontology[30]. 该数据集来自于数据库中最大的 4 个类,包括 Word、Sports、Business、Sci/Tec,有 1.2 万条样本. DBpedia Ontology 数据集是基于 Wikipedia 中最常用的信息盒(Infoboxes)由人工创建的,共有 56 万条数据.

3.4 模型参数设置

语言模型采用 Transformer 网络,模型包含 12 层解码器,并使用遮罩多头(masked multi-headed)注意力机制. 实验中设为 768 维隐藏层输出,12 个注意力头. 对于前馈网络,使用 3 072 维大小的向量. 优化方法使用 Adam 优化[26],最大学习率为 2.5×10^{-4}. Minibatch 大小设为 64. 由于在整个语言模型中广泛使用了 Layer Normalization[32],所以模型使用简单权重初始化.

3.5 实验结果

模型在数据集上取得的错误率如表 1 所示. 对于每个任务,选择了目前效果最好的模型进行比较. 在 IMDb 和 TREC-6 数据集上,与 CoVe(context vectors)模型[33]进行对比. 在 AG、Yelp 和 DBpedia 数据集上,与文献[34-35]提出的文本分类方法进行对比. 最后在全部数据集上与目前文本分类效果最好的基于微调的半监督模型 ULMFiT(universal language model fine-tuning)[2]进行对比.

表 1 文本分类实验错误率 %

模型	IMDb	Yelp-bi	Yelp-full	TREC-6	AG	DBpedia
CoVe[33]	8.2	–	–	4.2	–	–
Char-level CNN[34]	–	4.88	37.95	–	9.51	1.55
DPCNN[35]	–	2.64	30.58	–	6.87	0.88
ULMFiT[2]	4.6	2.16	29.98	3.6	5.01	0.80
PFTLM	4.3	1.95	27.86	3.4	4.88	0.78

在 IMDb 数据集上 PFTLM 模型的效果要好于 CoVe 模型,略好于 ULMFiT 模型. CoVe 模型是基于超列的转移学习方法. 这证明了不需要增加模型的复杂度,如加入各种注意力机制或者人工语义表示模型,基础的 Transformer 网络就能获得性能的提升. 这样能降低模型的复杂性,减少过拟合,更有利于扩展到其他的任务中. 与基于语言模型微调的方法 ULMFiT 对比,本文模型的效果略好于 ULMFiT 模型. 证明了 Transformer 的特征提取能力要好于 LSTM 模型,能捕获到更多、更远距离的语义信息. 首先,IMDb 数据集具有能代表真实数据集的特点,如文档长度、结构和邮件,在线评论等数据是否相似,这种类型的数据广泛存在于真实世界中;其次,情感分析任务也能运用到很多场景中,如商品反馈、舆情监控等实际任务. 总的来说,文本模型在 IMDb 模型上的性能可有效地迁移到其他实际场景中,而不需要进行大的改动.

在 Yelp-bi、Yelp-full、TREC-6、AG、DBpedia 这 5 个数据集上 PFILM 模型也取得了更好的性能. 在 TREC-6 数据集中,由于测试样本很少,只有 500 条,

PFTLM 模型的性能相较其他模型也有所提升，证明了 PFTLM 模型不仅适用于大数据量的数据集，在小数据量的数据集上也有好的表现.

4　结果分析

为了评估模型中各个部分的作用，设置了一系列的实验来证明. 选用 IMDb、TREC-6 和 AG 3 个具有代表性的数据集，分别代表不同的任务、特点和样本大小. 对于所有实验，采用 10% 的数据作为验证集. 每个分类器都训练 50 个循环.

1）预训练语言模型的作用. 比较了有预训练和没有预训练模型的效果，结果如表 2 所示. 可见，预训练对于在小数量数据集 TREC-6 上的性能有很大提升. 在实际的需求场景中，人工标注的成本十分昂贵，往往只有少量的标注样本，所以如何从少量的样本中学习到有用的信息是十分必要的. 通过预训练可以达到这样的效果. 同时，在数据量较为充足的数据集上性能也有一定的提升. 证明本文的方法同样适用于大型数据集.

表 2　是否包含预训练语言模型的错误率对比　%

预训练语言模型	IMDb	TREC-6	AG
不包含	5. 33	10. 06	5. 34
包含	4. 36	5. 26	4. 87

2）语言模型微调的作用. 没有微调和有微调的模型性能，对比结果如表 3 所示 . 其中微调包括上文提到的分层微调和斜三角学习速率. 微调语言模型在大型数据集 IMDb 的提高最为显著，在其他数据集上也有很好的表现.

表 3　不同语言模型微调的错误率对比　%

语言模型	IMDb	TREC-6	AG
不微调	7. 11	7. 58	6. 34
微调	4. 36	5. 26	4. 87

3）多探测任务微调的作用. 通过多探测任务微调和没有进行微调的模型对比可以看出，进行多探测任务微调的模型明显更好，结果如表 4 所示. 在数据量较小的 TREC-6 数据集上性能提升最为明显，证明了多探测任务微调有利于减少人工标注数据集的依赖.

除了多探测任务微调，用 Transformer 网络替代了 LSTM 网络，Transformer 具有比 LSTM 网络更好的远程语义抓取能力. Tai 等[36] 研究表明，Transformer 网络具有更高的并行计算能力和运行效率，更好的语义特征提取能力和长距离特征捕获能力等优势. 在 GPT[5] 的实验结果也表明，其他条件相同的情况下，Transformer 网络作为特征提取器的效果都要好于 LSTM 网络.

表 4　多探测任务微调模型的错误率对比　%

多探测任务	IMDb	TREC-6	AG
不微调	7. 19	8. 93	5. 25
微调	4. 36	5. 26	4. 87

实验结果证明，PFTLM 模型适用于广泛的文本分类任务，同时，语言模型加微调的方法在研究中可提供有效的帮助，如小语种的自然语言处理任务（有标注的训练集十分稀少）；新颖的自然语言处理任务（没有证明有效的神经网络模型）.

针对无监督的语言模型和针对特定任务的微调研究，未来会有更多的方法被提出. 笔者认为一个可能的方向是改进语言模型和微调，并提高可扩展性；另一个方向是将该方法应用于其他任务和模型. 对于一些形式更复杂的任务，如语义蕴含和智能问答等，可能需要不同于文本分类的方法来预训练和微调. 虽然对半监督学习的各个阶段都进行了分析，验证了其对模型性能的影响，但还需要研究预训练的语言模型捕获了哪些信息以及微调学习中对于不同任务所需的信息.

5　结束语

提出了基于探测任务的语言模型微调方法 PFTLM 用于文本分类模型. 这是一种有效的半监督学习方法，并证明了在文本分类任务上的有效性. 还提出了几种新颖的微调技术，这些技术可以防止信息被遗忘，并在各种文本分类任务中实现有效的信息迁移.

参考文献：

[1] Dai A M, Le Q V. Semi-supervised sequence learning [C] // Advances in Neural Information Processing Systems. Montréal, Canada: [s. n.], 2015: 3079-3087.

[2] Howard J, Ruder S. Universal language model fine-tuning for text classification[C] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2018: 328-339.

[3] Peters M, Ammar W, Bhagavatula C, et al. Semi-super-

vised sequence tagging with bidirectional language models [C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2017: 1756-1765.

[4] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[C]// Proceedings of NAACL-HLT. New Orleans: [s. n.], 2018: 2227-2237.

[5] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training [EB/OL]. (2018-06-11) [2019-06-17]. https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.

[6] Devlin J, Chang M, Lee K, et al. Bert: pre-training of deep bidirectional transformers for language understanding [EB/OL]. (2018-01-15) [2019-06-17]. https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_understanding_paper. pdf.

[7] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]// Advances in Neural Information Processing Systems. Long Beach, America: [s. n.], 2017: 5998-6008.

[8] Liu P J, Saleh M, Pot E, et al. Generating wikipedia by summarizing long sequences [C]// Sixth International Conference on Learning Representations Vancouver. Canada: [s. n.], 2018: 557-573.

[9] Kitaev N, Klein D. Constituencyparsing with a self-attentive encoder[C]// Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg. PA, USA: Association for Computational Linguistics, 2018: 2676-2686.

[10] Suzuki J, Isozaki H. Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data[C]// Proceedings of ACL-08: HLT. Columbus, Ohio: [s. n.], 2008: 665-673.

[11] Kamal N, Andrew M, Tom M. Semi-supervised text classification using EM[M]// Semi-Supervised Learning. Boston: The MIT Press, 2006: 32-55.

[12] Liang P. Semi-supervised learning for natural language [D]. Massachusetts: Massachusetts Institute of Technology, 2005.

[13] Chen Danqi, Manning C. A fast and accurate dependency parser using neural networks[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA: Association for Computational Linguistics, 2014: 740-750.

[14] Qi Ye, Sachan D, Felix M, et al. When and why are pre-trained word embeddings useful for neural machine translation? [C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Stroudsburg, PA, USA: Association for Computational Linguistics, 2018: 529-535.

[15] Logeswaran L, Lee H. An efficient framework for learning sentence representations [C]// Sixth International Conference on Learning Representations Vancouver. Canada: [s. n.], 2018: 1884-1891.

[16] Erhan D, Bengio Y, Courville A, et al. Why does unsupervised pre-training help deep learning? [J]. Journal of Machine Learning Research, 2010, 11(2): 625-660.

[17] Hochreiter S, Schmidhuber J. Long short-term memory [J]. NeuralComputation, 1997, 9(8): 1735-1780.

[18] Min S, Seo M, Hajishirzi H. Question answering through transfer learning from large fine-grained supervision data[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2017: 510-517.

[19] Severyn A, Moschitti A. UNITN: training deep convolutional neural network for twitter sentiment classification [C]// Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). Stroudsburg, PA, USA: Association for Computational Linguistics, 2015: 464-469.

[20] Sennrich R, Haddow B, Birch A. Improving neural machine translation models with monolingual data[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2016: 86-96.

[21] Mou Lili, Meng Zhao, Yan Rui, et al. How transferable are neural networks in NLP applications? [C]// Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016: 479-489.

[22] Adi Y, Kermany E, Belinkov Y, et al. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks [J]. arXiv preprint arXiv: 1608. 04207. 2016.

[23] Conneau A, Kruszewski G A N, Lample G, et al. What you can cram into a single vector: Probing sentence embeddings for linguistic properties[J]. CoRR, 2018: 01070.

[24] Linzen T, Dupoux E, Goldberg Y. Assessing the ability of LSTMs to learn syntax-sensitive dependencies[J]. Transactions of the Association for Computational Linguistics, 2016(4): 521-535.

[25] Gulordava K, Bojanowski P, Grave E, et al. Colorless green recurrent networks dream hierarchically[C]// Proceedings of NAACL-HLT. New Orleans, Louisiana: [s. n.], 2018: 1195-1205.

[26] Kingma D P, Ba J L. Adam: Amethod for stochastic optimization[C]// the 3rd International Conference on Learning Representations. San Diego, America: [s. n.], 2015: 351-365.

[27] Zhu Yukun, Kiros R, Zemel R, et al. Aligning books and movies: towards story-like visual explanations by watching movies and reading books[C]//2015 IEEE International Conference on Computer Vision (ICCV). New York: IEEE Press, 2015: 19-27.

[28] Yosinski J, Clune J, Bengio Y, et al. How transferable are features in deep neural networks? [C]// Advances in Neural Information Processing Systems. Montréal Canada: [s. n.], 2014: 3320-3328.

[29] Maas A L, Daly R E, Pham P T, et al. Learning word vectors for sentiment analysis[C]// Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon: [s. n.], 2011: 142-150.

[30] Zhang X, Zhao J, Lecun Y. Character-level convolutional networks for text classification[C]// Advances in Neural Information Processing Systems. Montréal Canada: [s. n.], 2015: 649-657.

[31] Voorhees E M, Tice D M. The TREC-8 question answering track evaluation[C]// TREC. Citeseer. Gaithersburg: [s. n.], 1999: 82.

[32] Ba J L, Kiros J R, Hinton G E. Layer normalization [J]. arXiv preprint arXiv: 1607. 06450. 2016.

[33] Mccann B, Bradbury J, Xiong C, et al. Learned in translation: contextualized word vectors[C]// Advances in Neural Information Processing Systems. Long Beach, USA: [s. n.], 2017: 6294-6305.

[34] Zhang Y, Wallace B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification[C]// Proceedings of the Eighth International Joint Conference on Natural Language Processing. Taipei: [s. n.], 2017: 253-263.

[35] Johnson R, Zhang Tong. Deep pyramid convolutional neural networks for text categorization[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2017: 562-570.

[36] Tai Kaisheng, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks[C]// Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2015: 1556-1566.