自然语言处理预训练模型综述

摘要：最近，预训练模型（PTM）*的出现将自然语言处理（NLP）带入了一个新时代。在本次调查中，我们对 NLP 的 PTM 进行了全面回顾。本文首先简要介绍了语言表征学习及其研究进展。然后，我们根据分类法从四个不同的角度对现有的 PTM 进行系统分类。接下来，我们将描述如何将 PTM 的知识应用于下游任务。最后，我们概述了 PTMs 未来研究的一些潜在方向。本调查旨在为理解、使用和开发各种 NLP 任务的 PTM 提供实践指南。

关键词：深度学习、神经网络、自然语言处理、预训练模型、分布式表示、单词嵌入、自监督学习、语言建模

1、 简介

随着深度学习的发展，各种神经网络被广泛用于解决自然语言处理（NLP）任务，如卷积神经网络（CNN）[1-3]、递归神经网络（RNN）[4,5]、基于图形的神经网络（GNN）[6-8]和注意机制[9,10]。这些神经模型的优点之一是能够缓解特征工程问题。非神经 NLP 方法通常严重依赖于离散的手工特征，而神经方法通常使用低维密集向量（又称分布式表示）来隐式表示语言的句法或语义特征。这些表述是在特定的 NLP 任务中学习的。因此，神经方法使人们很容易开发各种 NLP 系统。

尽管 NLP 任务的神经模型取得了成功，但与计算机视觉（CV）领域相比，性能改善可能不太显著。主要原因是，大多数受监督 NLP 任务的当前数据集非常小（机器翻译除外）。深度神经网络通常有大量的参数，这使得它们在这些小的训练数据上过拟合，在实践中不能很好地推广。因此，许多 NLP 任务的早期神经模型相对较浅，通常只有 1 个~3 个神经层。

近年来，大量研究表明，在大型语料库上预训练模型（PTM）可以学习通用语言表示，这有利于后续 NLP 任务，并且可以避免从头开始训练新模型。随着计算能力的发展，深层模型（即 Transformer[10]）的出现，以及训练技能的不断提高，PTMs 的体系结构已经从浅层向深层推进。第一代 PTM 旨在学习好的单词嵌入。因为下游任务不再需要这些模型本身，它们计算效率通常很低，例如 Skip Gram[11]和 GloV e[12]。

虽然这些经过预训练的嵌入可以捕获单词的语义，但它们与上下文无关，无法捕获上下文中更高层的概念，例如多义消歧、句法结构、语义角色、回指。第二代 PTM 侧重于学习上下文单词嵌入，如 CoV e[13]、ELMo[14]、OpenAI GPT[15]和 BERT[16]。这些学习过的编码器仍然需要通过下游任务在上下文中表示单词。此外，还提出了各种培训前任务，用于学习不同目的的 PTM。

本次调查的贡献总结如下：

1. 全面审查。我们全面回顾了 NLP 的 PTM，包括背景知识、模型体系结构、训练前任务、各种扩展、适应方法和应用。
2. 新分类法。我们提出了 NLP 的 PTM 分类法，从四个不同的角度对现有的 PTM 进行分类：1）表示类型，2）模型体系结构；3）培训前任务的类型；4） 针对特定场景类型的扩展。
3. 丰富的资源。我们收集了大量关于 PTMs 的资源，包括 PTMs 的开源实现、可视化工具、语料库和论文列表。
4. 未来的方向。我们讨论并分析了现有 PTM 的局限性。此外，我们还提出了未来可能的研究方向。

调查的其余部分组织如下。第 2 节概述了 PTM 的背景概念和常用符号。第 3 节简要概述了 PTM，并阐明了 PTM 的分类。第 4 节提供了 PTM 的扩展。第 5 节讨论了如何将 PTM 知识转移到下游任务中。第 6 节给出了 PTMs 的相关资源。第 7 节介绍了跨各种 NLP 任务的应用程序集合。第 8 节讨论了当前的挑战，并提出了未来的方向。第 9 节总结了本文。

2、 背景知识

2.1 语言表征学习

正如 Bengio 等人[17]所建议的那样，一个好的表征应该表达通用的先验知识，这些先验知识不是特定于任务的，但可能有助于学习机器解决人工智能任务。当涉及到语言时，一个好的表达应该捕捉隐藏在文本数据中的隐含语言规则和常识知识，例如词汇意义、句法结构、语义角色，甚至语用学。

分布式表示的核心思想是用低维实值向量来描述文本的意义。向量的每一个维度都没有对应的意义，而整体代表一个具体的概念。图 1 展示了 NLP 的通用神经架构。词的嵌入有两种：非语境嵌入和语境嵌入。

它们之间的区别在于，单词的嵌入是否会根据其出现的上下文而动态变化。

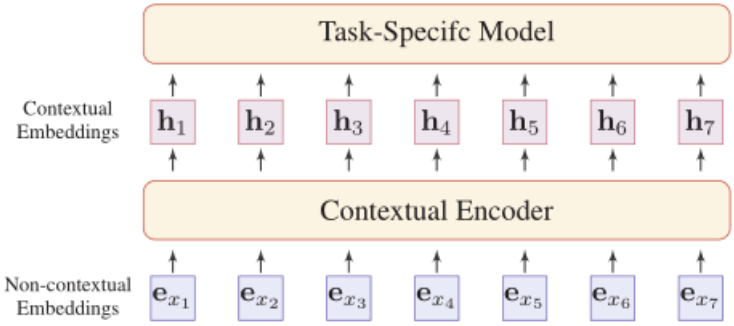

图 1：NLP 的通用神经架构

非语境嵌入：表示语言的第一步是将离散的语言符号映射到分布式嵌入空间。形式上，对于词汇表 V 中的每个单词（或子单词）x，我们将其映射到向量 ex∈RDe 带有查找表 E 的 RDe∈ RDe×| V |，其中 De 是一个超参数，指示令牌嵌入的维度。这些嵌入在任务数据和其他模型参数上进行训练。

这种嵌入有两个主要限制。第一个问题是嵌入是静态的。无论上下文如何，单词的嵌入都是一样的。因此，这些非语境嵌入无法模拟多义词。第二个问题是词汇表外的问题。为了解决这个问题，字符级单词表示或子单词表示被广泛应用于许多 NLP 任务中，例如 CharCNN[18]、FastText[19]和字节对编码（BPE）[20]。

语境嵌入：为了解决多义问题和单词的语境依赖性，我们需要区分不同语境中单词的语义。给定一个文本 x1，x2，⋯，xT，其中每个标记 ∈ V 是一个词或子词，xt 的上下文表示取决于整个文本。

$$[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_T] = f_{\text{enc}}(x_1, x_2, \cdots, x_T), \quad (1)$$

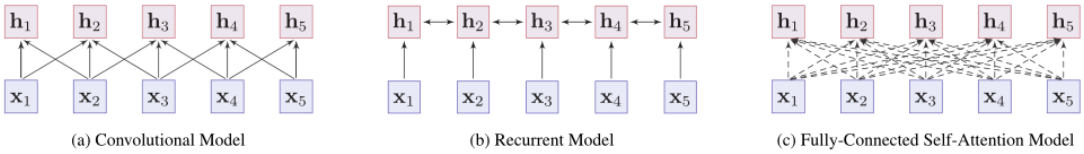

图 2：神经上下文编码器

其中 fenc（·）是神经编码器，如第 2.2 节所述，ht 称为上下文嵌入或动态 em-由于中包含的上下文信息，标记 xt 的层叠。

2.2 神经上下文编码器

大多数神经上下文编码器可分为两类：序列模型和非序列模型。图 2 展示了三种具有代表性的体系结构。

2.2.1 序列模型

序列模型通常按顺序捕捉单词的局部上下文。

卷积模型：卷积模型将单词嵌入到输入句子中，并通过卷积运算聚集来自相邻单词的局部信息来获取单词的含义[2]。

回归模型：循环模型捕捉了记忆较短的单词的上下文表示，如 LSTMs[21]和 GRUs[22]。实际上，双向 LSTM 或 GRU 用于从单词的两侧收集信息，但其性能往往受到长期依赖性问题的影响。

2.2.2 非序列模型

非序列模型通过词与词之间预定义的树或图形结构学习上下文表示，例如句法结构或语义关系。一些流行的非序列模型包括递归 NN[6]、TreeLSTM[7,23]和 GCN[24]。

尽管语言感知图结构可以提供有用的归纳偏差，但如何构建良好的图结构也是一个具有挑战性的问题。此外，该结构在很大程度上依赖于专家知识或外部 NLP 工具，例如依赖关系解析器。

全连接自我注意模型：在实践中，一种更直接的方法是使用完全连通的图来建模每两个单词之间的关系，并让模型自己学习结构。通常，连接权重是通过自我注意机制动态计算的，它隐含地表示单词之间的连接。完全连接的自我注意模型的一个成功例子是 Transformer[10,25]，它还需要其他补充模块，例如位置嵌入、层规范化、剩余连接和位置前馈网络（FFN）层。

2.2.3 分析

序列模型学习带有位置偏差的单词的上下文表示，很难捕捉单词之间的长期交互。然而，序列模型通常很容易训练，对于各种 NLP 任务都能得到很好的结果。

相比之下，作为一个实例化的全连接自我注意模型，Transformer 可以直接建模序列中每两个单词之间的依赖关系，这更强大，更适合建模语言的长期依赖关系。然而，由于其重型结构和较少的模型偏差，Transformer 通常需要大量的训练语料，并且容易在小型或中等规模的数据集上过度拟合[15,26]。

目前，变压器以其强大的容量成为 PTMs 的主流架构。

2.3 为什么进行预训练

随着深度学习的发展，模型参数的数量迅速增加。需要更大的数据集来全面训练模型参数并防止过度拟合。然而，对于大多数 NLP 任务来说，构建大规模标记数据集是一个巨大的挑战，因为注释成本非常昂贵，尤其是对于语法和语义相关的任务。

相比之下，大规模的未标记语料库相对容易构建。为了利用大量未标记的文本数据，我们可以首先从中学习良好的表示，然后将这些表示用于其他任务。最近的研究表明，借助于从大型未标注语料库的 PTM 中提取的表征，许多 NLP 任务的表现显著提高。

预训练的优点可以总结如下：

1. 在庞大的文本语料库上进行预训练可以学习通用语言表达和帮助完成下游任务。
2. 预训练提供了更好的模型初始化，这通常会导致更好的泛化性能，并加快对目标任务的收敛
3. 预训练可以被视为一种正则化，以避免在小数据上过度拟合[27]。

2.4 NLP 的 PTM 简史

预训练一直是学习深层神经网络参数的有效策略，然后对下游任务进行微调。早在 2006 年，深度学习的突破就出现了贪婪的分层无监督预训练，然后是监督微调[28]。在 CV 中，实际上是在庞大的 ImageNet 语料库上预训练模型，然后针对不同的任务对较小的数据进行进一步微调。这比随机初始化要好得多，因为模型学习一般的图像特征，这些特征可以用于各种视觉任务。

在 NLP 中，大规模语料库上的 PTM 也被证明有利于后续 NLP 任务，从浅层单词嵌入到深层神经模型。

2.4.1 第一代 PTM：预先培训的单词嵌入

将单词表示为密集向量有着悠久的历史[29]。神经网络语言模型（NNLM）[30]的先驱工作中引入了“现代”单词嵌入。Collobert 等人[31]表明，在未标记数据上预训练的单词嵌入可以显著改善许多 NLP 任务。为了解决计算复杂度问题，他们通过成对排序任务学习单词嵌入，而不是语言建模。他们的工作是首次尝试从未标记的数据中获得对其他任务有用的通用词嵌入。Mikolov 等人[11]表明，没有必要使用深层神经网络来构建良好的单词嵌入。他们提出了两种浅层结构：连续词包（CBOW）模型和 Skip-Gram（SG）模型。尽管简单，他们仍然可以学习高质量的单词嵌入，以捕捉单词之间潜在的句法和语义相

似性。Word2vec 是这些模型最流行的实现之一，它使预先训练好的单词嵌入可以用于 NLP 中的不同任务。此外，GloV e[12]也是一种广泛使用的用于获取预训练单词嵌入的模型，该模型由来自大型语料库的全局单词共现统计数据计算。

虽然预先训练的单词嵌入在 NLP 任务中已经被证明是有效的，但它们与上下文无关，并且大多由浅层模型训练。当用于下游任务时，整个模型的其余部分仍然需要从头开始学习。

在同一时期，许多研究人员还试图学习段落、句子或文档的嵌入，例如段落向量[32]、跳过思维向量[33]、上下文 2V ec[34]。与现代的继承者不同，这些句子嵌入模型试图将输入句子编码为固定维向量表示，而不是每个标记的上下文表示。

2.4.2 第二代 PTM：预先培训的上下文编码器

由于大多数 NLP 任务都超出了单词级别，所以在句子级别或更高级别上对神经编码器进行预训练是很自然的。神经编码器的输出向量也被称为上下文单词嵌入，因为它们根据上下文表示单词语义。

Dai 和 Le[35]提出了第一个成功的 NLP PTM 实例。他们用语言模型（LM）或序列自动编码器初始化 LSTM，发现预训练可以提高 LSTM 在许多文本分类任务中的训练和泛化能力。Liu 等人[5]用 LM 预先训练了一个共享的 LSTM 编码器，并在多任务学习（MTL）框架下对其进行了微调。他们发现，预训练和微调可以进一步提高 MTL 在多个文本分类任务中的性能。Ramachandran 等人[36]发现，通过无监督的预训练，Seq2Seq 模型可以显著改善。编码器和解码器的权值都用预先训练好的两种语言模型的权值初始化，然后用标记数据进行微调。除了使用 LM 对上下文编码器进行预训练外，McCann 等人[13]还使用机器翻译（MT）对从注意序列到序列模型的深层 LSTM 编码器进行了预训练。由预先训练的编码器输出的上下文向量（CoV e）可以提高各种常见 NLP 任务的性能。

由于这些早期的 PTM，现代 PTM 通常使用更大规模的语料库、更强大或更深层次的体系结构（如 Transformer）和新的培训前任务进行培训。

Peters 等人[14]使用双向语言模型（BiLM）预先训练的 2 层 LSTM 编码器，由前向 LM 和后向 LM 组成。经过预训练的 BiLM ELMo（语言模型嵌入）输出的上下文表示在广泛的 NLP 任务中带来了很大的改进。Akbik 等人[37]通过使用字符级 LM 预先训练的上下文字符串嵌入来捕捉单词的意思。然而，这两个 PTM 通常被用作特征提取器来生成

上下文单词嵌入，这些嵌入被输入到主模型中，用于下游任务。它们的参数是固定的，其余的都是固定的主要模型的参数仍然是从头开始训练的。ULMFiT（Universal Language Model Fine tuning）[38]试图对预先训练好的文本分类（TC）LM 进行微调，并在六个广泛使用的 TC 数据集上取得了最先进的结果。ULMFiT 包括 3 个阶段：1）关于一般领域数据的预培训；2） 对目标数据进行微调；3） 对目标任务进行微调。ULMFiT 还研究了一些有效的微调策略，包括区分微调、倾斜三角形学习率和逐渐解冻。

最近，非常深入的 PTM 在学习通用语言表示方面显示出了强大的能力：例如，OpenAI GPT（生成性预训练）[15]和 BERT（来自 Transformer 的双向编码器表示）[16]。除了 LM，还提出了越来越多的自我监督任务（见第 3.1 节），以使 PTM 从大规模文本语料库中获取更多知识。

自 ULMFiT 和 BERT 以来，微调已成为使 PTM 适应下游任务的主流方法。

3、 PTM 概述

PTM 之间的主要区别在于上下文编码器的使用、培训前任务和目的。在第 2.2 节中，我们简要介绍了上下文编码器的体系结构。在本节中，我们将重点介绍培训前任务，并对 PTM 进行分类。

3.1 预训练任务

训练前的任务对于学习语言的普遍表征至关重要。通常，这些培训前任务应该具有挑战性，并且有大量的培训数据。在本节中，我们将培训前任务总结为三类：监督学习、无监督学习和自我监督学习。

1. 监督学习（SL）是学习一个函数，该函数基于由输入-输出对组成的训练数据将输入映射到输出
2. 无监督学习（Unsupervised learning，UL）是从未标记的数据中发现一些内在知识，如聚类、密度、潜在表示。
3. 自我监督学习（SSL）是监督学习和非监督学习的混合体 1）。SSL 的学习模式与监督学习完全相同，但训练数据的标签是自动生成的。SSL 的关键思想是以某种形式预测来自其他部分的输入的任何部分。例如，蒙面语言模型（MLM）是

一个自我监督的任务，它试图在给定剩余单词的情况下预测句子中的蒙面单词。

在 CV 中，许多 PTM 在大型监督训练集（如 ImageNet）上进行训练。然而，在 NLP 中，大多数受监督任务的数据集都不足以训练好的 PTM。唯一的例外是机器翻译（MT）。大规模机器翻译数据集 WMT 2017 由 700 多万个句子对组成。此外，机器翻译是 NLP 中最具挑战性的任务之一，在机器翻译上预先训练的编码器可以使各种下游 NLP 任务受益。作为一个成功的 PTM，CoV e[13]是一个在机器翻译任务上预先训练过的编码器，它改进了多种常见的 NLP 任务：情绪分析（SST、IMDb）、问题分类（TREC）、蕴涵（SNLI）和问题回答（SQuAD）。

在本节中，我们将介绍一些在现有 PTM 中广泛使用的培训前任务。我们可以将这些任务视为自我监督学习。表 1 还总结了它们的损失函数。

3.1.1 语言建模（LM）

NLP 中最常见的无监督任务是概率语言建模（LM），这是一个经典的概率密度估计问题。虽然 LM 是一个通用概念，但在实践中，LM 通常特别指自回归 LM 或单向 LM。

给定一个文本序列 x1:T=[x1，x2，··，xT]，其联合概率 p（x1:T）可以分解为

$$p(\mathbf{x}_{1:T}) = \prod_{t=1}^{T} p(x_t|\mathbf{x}_{0:t-1}), \qquad (2)$$

其中 x0 是表示序列开始的特殊标记。

条件概率 p（xt | x0:t）−1） 可以通过给定语言上下文 x0:t 的词汇概率分布来建模−1.上下文 x0:t−1 由神经编码器 fenc（·）建模，条件概率为

$$p(x_t|\mathbf{x}_{0:t-1}) = g_{\mathrm{LM}}\left(f_{\mathrm{enc}}(\mathbf{x}_{0:t-1})\right), \qquad (3)$$

其中 gLM（·）是预测层。

给定一个庞大的语料库，我们可以用最大似然估计（MLE）来训练整个网络。

单向 LM 的一个缺点是，每个标记的表示只对左向上下文标记及其本身进行编码。然而，更好的文本上下文表示应该从两个方向对上下文信息进行编码。

Table 1: Loss Functions of Pre-training Tasks

Task	Loss Function	Description
LM	$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{T} \log p(x_t \mid \mathbf{x}_{<t})$	$\mathbf{x}_{<t} = x_1, x_2, \cdots, x_{t-1}$.
MLM	$\mathcal{L}_{\text{MLM}} = -\sum_{\hat{x} \in m(\mathbf{x})} \log p\left(\hat{x} \mid \mathbf{x}_{\backslash m(\mathbf{x})}\right)$	$m(\mathbf{x})$ and $\mathbf{x}_{\backslash m(\mathbf{x})}$ denote the masked words from $\mathbf{x}$ and the rest words respectively.
Seq2Seq MLM	$\mathcal{L}_{\text{S2SMLM}} = -\sum_{t=i}^{j} \log p\left(x_t \mid \mathbf{x}_{\backslash \mathbf{x}_{i:j}}, \mathbf{x}_{i:t-1}\right)$	$\mathbf{x}_{i:j}$ denotes an masked n-gram span from i to j in $\mathbf{x}$.
PLM	$\mathcal{L}_{\text{PLM}} = -\sum_{t=1}^{T} \log p(z_t \mid \mathbf{z}_{<t})$	$\mathbf{z} = perm(\mathbf{x})$ is a permutation of $\mathbf{x}$ with random order.
DAE	$\mathcal{L}_{\text{DAE}} = -\sum_{t=1}^{T} \log p(x_t \mid \hat{\mathbf{x}}, \mathbf{x}_{<t})$	$\hat{\mathbf{x}}$ is randomly perturbed text from $\mathbf{x}$.
DIM	$\mathcal{L}_{\text{DIM}} = s(\hat{\mathbf{x}}_{i:j}, \mathbf{x}_{i:j}) - \log \sum_{\bar{\mathbf{x}}_{i:j} \in \mathcal{N}} s(\hat{\mathbf{x}}_{i:j}, \bar{\mathbf{x}}_{i:j})$	$\mathbf{x}_{i:j}$ denotes an n-gram span from i to j in $\mathbf{x}$, $\hat{\mathbf{x}}_{i:j}$ denotes a sentence masked at position i to j, and $\bar{\mathbf{x}}_{i:j}$ denotes a randomly-sampled negative n-gram from corpus.
NSP/SOP	$\mathcal{L}_{\text{NSP/SOP}} = -\log p(t \mid \mathbf{x}, \mathbf{y})$	$t = 1$ if $\mathbf{x}$ and $\mathbf{y}$ are continuous segments from corpus.
RTD	$\mathcal{L}_{\text{RTD}} = -\sum_{t=1}^{T} \log p(y_t \mid \hat{\mathbf{x}})$	$y_t = \mathbf{1}(\hat{x}_t = x_t)$, $\hat{\mathbf{x}}$ is corrupted from $\mathbf{x}$.

[1] $\mathbf{x} = [x_1, x_2, \cdots, x_T]$ denotes a sequence.

一种改进的解决方案是双向 LM（BiLM），它由两个单向 LM 组成：一个向前的从左到右 LM 和一个向后的从右到左 LM。对于 BiLM，Baevski 等人[39]提出了一个双塔模型，即前塔操作从左到右的 LM，后塔操作从右到左的 LM。

3.1.2 蒙面语言建模（MLM）

蒙面语言建模（MLM）最早由 Taylor[40]在文献中提出，他将其称为完形填空任务。Devlin 等人[16]将该任务改编为一项新的训练前任务，以克服标准单向 LM 的缺点。不严格地说，MLM 首先从输入句子中屏蔽掉一些标记，然后训练模型，用其他标记预测屏蔽的标记。然而，这种预训练方法将在预训练阶段和微调阶段之间产生不匹配，因为掩码标记在微调阶段不会出现。根据经验，为了解决这个问题，Devlin 等人[16]在 80%的时间里使用一个特殊的[MASK]标记，在 10%的时间里使用一个随机标记，在 10%的时间里使用原始标记来执行掩蔽。

序列间传销（Seq2Seq 传销）：传销通常作为分类问题来解决。我们将遮罩序列馈送给神经编码器，其输出向量进一步馈送到 softmax 分类器以预测遮罩令牌。或者，我们可以为 MLM 使用编码器-解码器（又名 sequence-tosequence）架构，在该架构中，编码器被馈送一个屏蔽序列，并且解码器以自动回归的方式顺序产生屏蔽令牌。我们将这种传销称为序列对序列传销（Seq2Seq MLM），用于 MASS[41]和 T5[42]。Seq2Seq MLM 可以使 Seq2Seq 风格的下游任务受益，例如问答、摘要和机器翻译。

增强的蒙面语言建模（E-MLM）：同时，有多个研究提出了不同的传销增强版本，以进一步改善伯特。RoBERTa[43]通过动态掩蔽改进了BERT，而不是静态掩蔽。

UniLM[44,45]将掩码预测任务扩展到三种语言建模任务：单向、双向和序列到序列预测。XLM[46]对一组并行双语句子对进行传销，称为翻译语言建模（Translation Language Modeling，TLM）。SpanBERT[47]用随机连续词掩蔽和跨距边界目标（SBO）代替 MLM，将结构信息整合到预训练中，这要求系统根据跨距边界预测掩蔽跨距。此外，StructBERT[48]引入了跨度顺序恢复任务，以进一步整合语言结构。

另一种丰富传销的方法是吸收外部知识（见第 4.1 节）。

3.1.3 置换语言建模（PLM）

尽管传销任务在培训前被广泛使用，Yang 等人[49]声称，当模型应用于下游任务时，传销预培训中使用的一些特殊标记，如[MASK]，是不存在的，这导致了预培训和微调之间的差距。为了克服这个问题，置换语言建模（PLM）[49]是一个培训前目标尽管传销任务在培训前被广泛使用，Yang 等人[49]声称，当模型应用于下游任务时，传销预培训中使用的一些特殊标记，如[MASK]，是不存在的，这导致了预培训和微调之间的差距。为了克服这个问题，置换语言建模（PLM）[49]是一个与训练目标。

3.1.4 去噪自动编码器（DAE）

去噪自动编码器（DAE）接收部分损坏的输入，旨在恢复原始未失真的输入。特定于语言，使用序列到序列模型（如标准转换器）来重建原始文本。有几种方法可以破坏文本[50]：

（1） 令牌屏蔽：从输入中随机抽取令牌，并将其替换为[MASK]元素。

（2） 令牌删除：从输入中随机删除令牌。与令牌屏蔽不同，该模型需要确定缺失输入的位置。

（3） 文本填充：与 SpanBERT 一样，对多个文本跨距进行采样，并用单个[MASK]标记替换。每个跨距长度由泊松分布（λ=3）得出。该模型需要预测一个跨度中缺少多少标记。

（4） 句子排列：根据句号将文档分成句子，并随机排列这些句子。

（5） 文档旋转：随机均匀地选择一个标记，并旋转文档，使其以该标记开始。模型需要确定文档的实际起始位置。

3.1.5 对比学习（CTL）

对比学习[51]假设一些观察到的文本在语义上比随机抽样的文本更相似。学习文本对（x，y）的得分函数 s（x，y），以最小化目标函数：

$$\mathcal{L}_{\mathrm{CTL}} = \mathbb{E}_{x,y^+,y^-}\Big[-\log \frac{\exp\left(s(x,y^+)\right)}{\exp\left(s(x,y^+)\right)+\exp\left(s(x,y^-)\right)}\Big], \qquad (4)$$

其中（x，y+）是一对相似的线对，y− 可能与 x.y+和 y 不同− 通常称为阳性和阴性样本。分数函数 s（x，y）通常由可学习的神经编码器以两种方式计算：s（x，y）=f Tenc（x）fenc（y）或 s（x，y）=fenc（x）⊕ y）。

CTL 背后的理念是“通过比较学习”。与 LM 相比，CTL 通常具有较少的计算复杂度，因此是 PTM 的理想替代训练标准。

Collobert 等人[31]提出了区分真假短语的成对排序任务。该模型需要预测一个法律短语的得分高于用随机词替换中心词得到的错误短语的得分。Mnih 和 Kavukcuoglu[52]使用噪声对比估计（NCE）[53]有效地训练单词嵌入，该方法训练一个二元分类器来区分真假样本。NCE 的思想也用于著名的 word2vec 嵌入[11]。

在下面的段落中，我们将简要介绍一些最近提出的 CTL 任务。

深度信息 Max（昏暗）： Deep InfoMax（DIM）[54]最初是针对图像提出的，它通过最大化图像表示和图像局部区域之间的互信息来提高表示的质量。

Kong 等人[55]将 DIM 应用于语言表征学习。序列 x 的全局表示被定义为上下文编码器 fenc（x）输出的第一个标记（假定为句子的特殊起始符号）的隐藏状态。DIM 的目标是为 fenc（xi:j）T fenc（ˆxi:j）分配比 fenc（ˆxi:j）T fenc（ˆxi:j）更高的分数，其中 xi:j 表示 x 中从 i 到 j 的 n-gram2）跨度，ˆxi:j 表示在位置 i 到 j 处隐藏的句子，并且 ˆxi:j 表示从语料库中随机抽样的负 n-gram。

替换令牌检测（RTD）： 替换令牌检测（RTD）与 NCE 相同，但根据周围环境预测是否替换令牌。

带有否定抽样的 CBOW（CBOW-NS）[11]可以被视为 RTD 的一个简单版本，其中否定样本是从具有简单提议分布的词汇表中随机抽样的。

ELECTRA[56]通过使用生成器替换序列中的一些令牌来改进 RTD。发生器 G 和鉴别器 D 按照两个阶段的程序进行培训：（1）仅对具有传

销任务的发生器进行 n1 步培训；（2） 用生成器的权重初始化鉴别器的权重。然后训练鉴别器进行 n2 步的辨别任务，保持 G 冻结。在这里，区别性任务指示是否将输入标记替换为 G。生成器在预训练后抛出，只有鉴别器会在下游任务上进行微调。

RTD 也是不匹配问题的替代解决方案。网络在训练前会看到[屏蔽]，但在下游任务中进行微调时不会看到。

类似地，WKLM[57]取代了实体级而非标记级的单词。具体地说，WKLM 用相同类型的其他实体的名称替换实体提及，并训练模型来区分实体是否已被替换。

下一句预测（NSP）：标点符号是文本数据的自然分隔符。因此，利用这些方法构建预训练方法是合理的。下一句预测（NSP）[16]就是一个很好的例子。顾名思义，NSP 训练模型来区分两个输入句子是否是训练语料库中的连续片段。具体来说，当为每个预训练示例选择句子对时，50%的时候，第二个句子是第一个示例的实际下一个句子，50%的时候，它是语料库中的随机句子。通过这样做，它能够教会模型理解两个输入句子之间的关系，从而有利于对这些信息敏感的下游任务，如问答和自然语言推理。

然而，NSP 任务的必要性受到后续工作的质疑[47、49、43、63]。Yang 等人[49]发现 NSP 任务的影响不可靠，而 Joshi 等人[47]发现没有 NSP 损失的单句训练优于有 NSP 损失的句子对训练。此外，Liu 等人[43]对 NSP 任务进行了进一步的分析，结果表明，当使用单个文档中的文本块进行训练时，删除 NSP 丢失匹配项或稍微提高下游任务的性能。

句子顺序预测（SOP）：为了更好地模拟句间连贯性，ALBERT[63]将 NSP 损失替换为句子顺序预测（SOP）损失。正如 Lan 等人[63]推测的那样，NSP 将话题预测和连贯性预测融合在一个任务中。因此，该模型可以仅依靠更简单的任务，即主题预测来进行预测。与 NSP 不同，SOP 使用同一文件中的两个连续段作为正面示例，使用相同的两个连续段，但顺序交换为负面示例。因此，在各种下游任务上，ALBERT 始终优于 BERT。StructBERT[48]和 BERTje[88]也将 SOP 作为他们的自我监督学习任务。

3.1.6 其他

除上述任务外，还有许多其他辅助性的培训前任务，用于整合事实知识（见第 4.1 节）、改进跨语言任务（见第 4.2 节）、多模式应用（见第 4.3 节）或其他特定任务（见第 4.4 节）。

3.2 PTM 的分类

为了阐明 NLP 中现有 PTM 之间的关系，我们建立了 PTM 分类法，从四个不同的角度对现有 PTM 进行分类：

1. 表示类型：根据下游任务使用的表示，我们可以将 PTM 分为非上下文模型和上下文模型。
2. 架构：PTMs 使用的主干网，包括 LSTM、变压器编码器、变压器解码器和全变压器架构。“Transformer”指标准编码器-解码器架构。“变压器编码器”和“变压器解码器”分别指标准变压器架构的编码器和解码器部分。它们的不同之处在于，解码部分使用带三角形矩阵的掩蔽自我注意，以防止令牌进入其未来（右）位置。
3. 培训前任务类型：PTM 使用的培训前任务类型。我们已在第 3.1 节中讨论过这些问题。
4. 扩展：为各种场景设计的 PTM，包括知识丰富的 PTM、多语言或特定语言的 PTM、多模型 PTM、特定领域的 PTM 和压缩的 PTM。我们将在第 4 节中详细介绍这些扩展。

图 3 显示了分类法以及一些相应的有代表性的 PTM。此外，表 2 更详细地区分了一些有代表性的 PTM。

3.3 模型分析

由于 PTMs 的巨大成功，了解 PTMs 获取了哪些知识以及如何从中归纳知识非常重要。有大量文献分析语言知识和世界知识，这些知识存储在预先训练好的非语境和语境嵌入中。

3.3.1 非上下文嵌入

静态单词嵌入首先是探索各种知识。Mikolov 等人[117]发现，通过神经网络语言模型学习的单词表征能够捕捉语言中的语言规律，单词之间的关系可以通过特定于关系的向量偏移来表征。进一步的类比实验[11]表明，skip-gram 模型产生的词向量

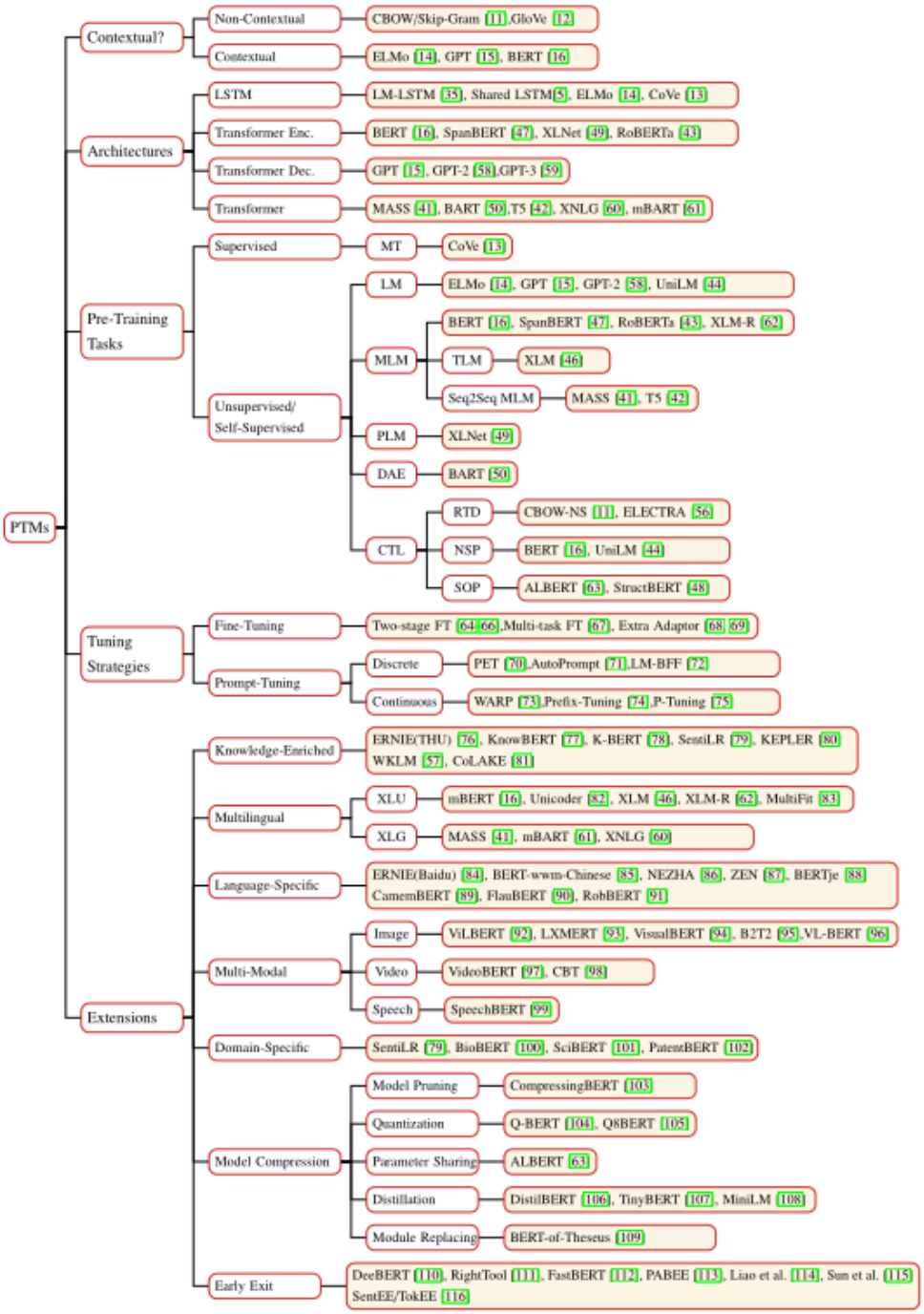

Figure 3: Taxonomy of PTMs with Representative Examples

Table 2: List of Representative PTMs

PTMs	Architecture†	Input	Pre-Training Task	Corpus	Params	GLUE‡	FT?¶
ELMo [14]	LSTM	Text	BiLM	WikiText-103			No
GPT [15]	Transformer Dec.	Text	LM	BookCorpus	117M	72.8	Yes
GPT-2 [58]	Transformer Dec.	Text	LM	WebText	117M ~ 1542M		No
BERT [16]	Transformer Enc.	Text	MLM & NSP	WikiEn+BookCorpus	110M ~ 340M	81.9*	Yes
InfoWord [55]	Transformer Enc.	Text	DIM+MLM	WikiEn+BookCorpus	=BERT	81.1*	Yes
RoBERTa [43]	Transformer Enc.	Text	MLM	BookCorpus+CC-News+OpenWebText+ STORIES	355M	88.5	Yes
XLNet [49]	Two-Stream Transformer Enc.	Text	PLM	WikiEn+ BookCorpus+Giga5 +ClueWeb+Common Crawl	≈BERT	90.5§	Yes
ELECTRA [56]	Transformer Enc.	Text	RTD+MLM	same to XLNet	335M	88.6	Yes
UniLM [44]	Transformer Enc.	Text	MLM+ NSP	WikiEn+BookCorpus	340M	80.8	Yes
MASS [41]	Transformer	Text	Seq2Seq MLM	*Task-dependent			Yes
BART [50]	Transformer	Text	DAE	same to RoBERTa	110% of BERT	88.4*	Yes
T5 [42]	Transformer	Text	Seq2Seq MLM	Colossal Clean Crawled Corpus (C4)	220M ~ 11B	89.7*	Yes
ERNIE(THU) [76]	Transformer Enc.	Text+Entities	MLM+NSP+dEA	WikiEn + Wikidata	114M	79.6	Yes
KnowBERT [77]	Transformer Enc.	Text	MLM+NSP+EL	WikiEn + WordNet/Wiki	253M ~ 523M		Yes
K-BERT [78]	Transformer Enc.	Text+Triples	MLM+NSP	WikiZh + WebtextZh + CN-DBpedia + HowNet + MedicalKG	=BERT		Yes
KEPLER [80]	Transformer Enc.	Text	MLM+KE	WikiEn + Wikidata/WordNet			Yes
WKLM [57]	Transformer Enc.	Text	MLM+ERD	WikiEn + Wikidata	=BERT		Yes
CoLAKE [81]	Transformer Enc.	Text+Triples	MLM	WikiEn + Wikidata	=RoBERTa	86.3	Yes

† "Transformer Enc." and "Transformer Dec." mean the encoder and decoder part of the standard Transformer architecture respectively. Their difference is that the decoder part uses masked self-attention with triangular matrix to prevent tokens from attending their future (right) positions. "Transformer" means the standard encoder-decoder architecture.
‡ the averaged score on 9 tasks of GLUE benchmark (see Section 7.1).
* without WNLI task.
§ indicates ensemble result.
¶ means whether is model usually used in fine-tuning fashion.
◦ The MLM of UniLM is built on three versions of LMs: Unidirectional LM, Bidirectional LM, and Sequence-to-Sequence LM.

可以捕捉句法和语义词的关系，例如 vec（“中国”）− vec（“北京”）≈ vec（“日本”）− vec（“东京”）。此外，他们还发现了词向量的组成特性，例如，vec（“德语”）+vec（“大写”）与 vec（“柏林”）很接近。受这些工作的启发，Rubinstein 等人[118]发现，分布词表征能够很好地预测分类属性（例如狗是动物），但无法学习定语属性（例如天鹅是白色的）。类似地，Gupta 等人[119]表明 word2vec 嵌入隐式编码实体的引用属性。分布式词向量，以及一个简单的监督模型，可以学习以合理的准确度预测实体的数字和二进制属性。

3.3.2 上下文嵌入

大量研究探索并归纳了语境嵌入中不同类型的知识。一般来说，有两种类型的知识：语言知识和世界知识。

语言知识：研究人员设计了一系列广泛的探究任务来研究 PTMs 中的语言知识。Tenney 等人[120]，Liu 等人[121]发现 BERT 在许多句法任务上表现良好，比如词性标注和成分标注。然而，与简单的句法任务相比，伯特在语义和细粒度句法任务方面做得不够好。

此外，Tenney 等人[122]分析了 BERT 层在不同任务中的作用，发现 BERT 以类似于 NLP 管道中的顺序解决任务。此外，关于主谓一致[123]和语义角色[124]的知识也被证实存在于 BERT 中。此外，Hewitt 和 Manning[125]、Jawahar 等人[126]、Kim 等人[127]提出了几种从 BERT 中提取依赖树和选区树的方法，证明了 BERT 对语法结构进行编码的能力。Reif 等人[128]研究了 BERT 中内部表征的几何结构，并找

到了一些证据：1）语言特征似乎表现在单独的语义和句法子空间中；2） 注意矩阵包含语法表征；3） 伯特能很好地辨别词义。

世界知识：除了语言知识，PTMs 还可以存储训练数据中呈现的世界知识。探索世界知识的一种简单方法是用“填空”完形填空语句来查询伯特，例如，“但丁出生于[面具]”。Petroni 等人[129]通过从多个知识源手动创建单标记完形填空语句（查询），构建了 LAMA（语言模型分析）任务。他们的实验表明，BERT 所包含的世界知识与传统知识具有竞争力信息提取方法。由于 LAMA 中查询生成过程的简单性，Jiang 等人[130]认为，LAMA 只是测量了语言模型所知道的下限，并提出了更高级的方法来生成更高效的查询。尽管喇嘛的发现令人惊讶，但它也受到了后续研究的质疑[131132]。类似地，一些研究从 BERT 中归纳出下游任务的关系知识[133]和常识知识[134]。

4、 PTM 的扩展

4.1 知识丰富的 PTM

PTM 通常从通用的大规模文本语料库中学习通用语言表示，但缺乏特定领域的知识。将外部知识库中的领域知识整合到 PTM 中已被证明是有效的。外部知识的范围从语言[135,79,77,136]、语义[137]、常识[138]、事实[76-78,57,80]到领域特定知识[139,78]。

一方面，外部知识可以在培训前注入。早期的研究[140–143]集中于联合学习知识图嵌入和单词嵌入。自 BERT 以来，一些辅助培训前任务旨在将外部知识融入深度 PTM。LIBERT[135]（语言知识型 BERT）通过额外的语言约束任务整合了语言知识。Ke 等人[79]整合了每个词的情感极性，将传销扩展到标签意识传销（LA-MLM）。因此，他们提出的模型 SentiLR 在几个句子和体层情绪分类任务上实现了最先进的性能。Levine 等人[137]提出了 SenseBERT，它经过预先训练，不仅可以预测蒙面代币，还可以预测它们在 WordNet 中的超值。ERNIE（THU）[76]将预先训练在知识图上的实体嵌入与文本中相应的实体提及相结合，以增强文本表示。类似地，KnowBERT[77]通过实体链接模型联合训练 BERT，以端到端的方式合并实体表示。Wang 等人[80]提出了开普勒，它联合优化了知识嵌入和语言建模目标。这些工作通过实体嵌入来注入知识图的结构信息。相比之下，K-BERT[78]明确地将从 KG 中提取的相关三元组注入到句子中，以获得 BERT 的扩展树形输入。CoLAKE[81]将知识语

境和语言语境整合成一个统一的图形，然后用 MLM 对其进行预训练，以获得知识和语言的语境化表示。此外，Xiong 等人[57]采用了实体替换识别，以鼓励模型更加了解事实知识。然而，这些方法大多在注入知识时更新 PTMs 的参数，在注入多种知识时可能会出现灾难性遗忘。为了解决这个问题，K-Adapter[136]通过为不同的培训前任务独立培训不同的适配器，注入了多种知识，从而允许持续的知识注入。

另一方面，可以将外部知识整合到预先训练的模型中，而无需从头开始重新训练它们。例如，K-BERT[78]允许在对下游任务进行微调时注入事实知识。Guan 等人[138]使用常识性知识库 ConceptNet 和 TOMIC 来增强故事生成的 GPT-2。Yang 等人[144]提出了一个知识-文本融合模型，以获取机器阅读理解中的相关语言和事实知识。

此外，Logan IV 等人[145]和 Hayashi 等人[146]分别将语言模型扩展为知识图语言模型（KGLM）和潜在关系语言模型（LRLM），这两种模型都允许以知识图为条件进行预测。这些新颖的 KG 条件语言模型显示了预训练的潜力。

4.2 多语言和特定语言的 PTM

4.2.1 多语言 PTM

学习跨语言共享的多语言文本表示在许多跨语言 NLP 任务中起着重要作用。

跨语言理解（XLU）：早期的大部分工作都集中在学习多语言单词嵌入[147–149]，它在一个语义空间中表示来自多种语言的文本。然而，这些方法通常需要（弱）语言之间的对齐。

多语种 BERT3（mBERT）由传销公司进行预培训，在维基百科文本上分享前 104 种语言的词汇和权重。每个培训样本都是单语文档，没有专门设计的跨语言目标，也没有任何跨语言数据。即便如此，姆伯特在跨语言概括方面的表现也出人意料地出色[150]。K 等人[151]表明，语言之间的词汇重叠在跨语言成功中起着微不足道的作用。

XLM[46]通过合并跨语言任务翻译语言建模（translation language modeling，TLM）改进了 mBERT，该任务在并行双语句子对的串联上执行 MLM。Unicoder[82]进一步提出了三个新的跨语言预训练任务，包括跨语言单词恢复、跨语言释义分类和分类屏蔽语言模型（XMLM）。

XLM RoBERTa（XLM-R）[62]是一款可缩放的多语言编码器，它根据大量的训练数据进行预训练，使用 100 种不同语言的 2.5TB 数

据。XLM RoBERTa 的培训前任务仅限于单语传销。XLM-R 在多个跨语言基准（包括 XNLI、MLQA 和 NER）上取得了最新成果。

跨语言生成（XLG）：多语言生成是一种从输入语言生成不同语言文本的任务，例如机器翻译和跨语言抽象摘要。

与用于多语言分类的 PTM 不同，用于多语言生成的 PTM 通常需要同时对编码器和解码器进行预训练，而不是只关注编码器。

MASS[41]使用单语 Seq2Seq MLM 在多种语言上预先训练 Seq2Seq 模型，并在无监督 NMT 方面取得显著改进。XNLG[60]为跨语言自然语言生成执行两阶段预训练。第一阶段用单语 MLM 和跨语言 MLM（XMLM）任务对编码器进行预训练。第二阶段通过使用单语 DAE 和跨语言自动编码（XAE）任务预先训练解码器，同时保持编码器固定。实验表明，XNLG 在跨语言问题生成和跨语言抽象摘要方面具有优势。mBART[61]是 BART[50]的多语言扩展，它与 Seq2Seq 去噪自动编码器（DAE）任务一起，对 25 种语言的大规模单语语料库进行预训练。实验表明，mBART 在各种机器翻译（MT）任务中都能显著提高性能。

4.2.2 特定语言的 PTM

尽管多语言 PTM 在许多语言上表现良好，但最近的研究表明，在单一语言上训练的 PTM 显著优于多语言结果[89,90152]。

对于没有明确单词边界的汉语，对更大粒度[85,87,86]和多粒度[84,153]的单词表示进行建模已显示出巨大的成功。库拉托夫和阿尔希波夫[154]使用迁移学习技术将多语言的 PTM 调整为俄语的单语 PTM。此外，针对不同语言发布了一些单语 PTM，如法语的 CamemBERT[89]和福楼拜[90]，芬兰语的 FinBERT[152]，荷兰语的 BERTje[88]和 RobBERT[91]，阿拉伯语的 AraBERT[155]。

4.3 多模式 PTM

通过观察 PTMs 在许多 NLP 任务中的成功，一些研究集中于获得 PTMs 的跨模态版本。这些模型中的绝大多数是为一般的视觉和语言特征编码而设计的。这些模型是在一些巨大的跨模态数据语料库上进行预训练的，例如带有口语的视频或带有字幕的图像，并结合扩展的预训练任务，以充分利用多模态特征。通常，基于视觉的传销、蒙面视觉特征建模和视觉语言匹配等任务广泛应用于多模态预训练中，如 VideoBERT[97]、VisualBERT[94]、ViLBERT[92]。

4.3.1 视频文本 PTM

VideoBERT[97]和 CBT[98]是视频和文本的联合模型。为了获得用于预训练的视觉和语言标记序列，视频分别通过基于 CNN 的编码器和现成的语音识别技术进行预处理。在处理后的数据上训练单变压器编码器，学习视频字幕等下游任务的视觉语言表示。此外，UniViLM[156]建议引入生成任务，以进一步预训练在下游任务中使用的解码器。

4.3.2 图像文本 PTMs

除了视频语言预训练的方法外，还有几项工作在图像-文本对上引入了 PTM，旨在适应视觉问答（VQA）和视觉常识推理（VCR）等下游任务。一些提出的模型采用两个单独的编码器分别表示图像和文本，如 ViLBERT[92]和 LXMERT[93]。而 VisualBERT[94]、B2T2[95]、VLBERT[96]、Unicoder VL[157]和 UNITER[158]等其他方法则提出了单流统一变压器。虽然这些模型结构不同，但在这些方法中引入了类似的预训练任务，如传销和图像文本匹配。为了更好地利用视觉元素，在使用预先训练好的变换器进行编码之前，通过应用 RoI 或边界框检索技术将图像转换为区域序列。

4.3.3 音频文本 PTM

此外，有几种方法探索了在音频-文本对上使用 PTM 的可能性，例如 SpeechBERT[99]。本文试图通过使用单个转换器编码器对音频和文本进行编码，建立一个端到端语音问答（SQA）模型，该编码器在语音和文本语料库上使用 MLM 进行预训练，并在问答过程中进行微调。

4.4 特定领域和特定任务的 PTM

大多数公开的 PTM 都是在维基百科等通用领域语料库上进行培训的，这将其应用程序限制在特定领域或任务上。最近，一些研究提出了在专业语料库上训练 PTM 的建议，例如 BioBERT[100]用于生物医学文本，SciBERT[101]用于科学文本，ClinicalBERT[159，160]用于临床文本。

除了对特定领域的 PTM 进行预训练外，一些工作还试图使可用的预训练模型适应目标应用，例如生物医学实体规范化[161]、专利分类[102]、进度注释分类和关键字提取[162]。还提出了一些面向任务的预训练任务，例如用于情感分析的 SentiLR[79]中的情感标签感知传销，用于文本摘要的间隙句子生成（GSG）[163]，以及用于不流畅检测的噪声词检测[164]。

4.5 模型压缩

由于 PTM 通常由至少数亿个参数组成，它们很难部署在现实应用中的在线服务和资源受限的设备上。模型压缩[165]是减少模型大小和提高计算效率的潜在方法。

压缩 PTM 的方法有五种[166]：（1）模型修剪，删除不太重要的参数；（2）权重量化[167]，使用更少的位来表示参数；（3）在相似的模型单元之间共享参数；（4）知识提取[168]，它训练一个较小的学生模型，从原始模型的中间输出中学习；（5）模块替换，它用更紧凑的替代品取代了原有 PTM 的模块。

表 3 给出了一些有代表性的压缩 PTM 的比较。

4.5.1 模型修剪

模型修剪是指去除部分神经网络（例如权重、神经元、层、通道、注意头），从而达到减小模型大小和加快推理时间的效果。

Gordon 等人[103]探讨了修剪的时机（例如，在训练前、下游微调后进行修剪）和修剪制度。Michel 等人[174]和 V oita 等人[175]试图剪掉变压器组中所有的自我注意头。

4.5.2 量化

量化是指将精度较高的参数压缩为精度较低的参数。Shen 等人[104]和 Zafrir 等人[105]的作品仅关注这一领域。请注意，量化通常需要兼容的硬件。

4.5.3 参数共享

另一种减少参数数量的众所周知的方法是参数共享，广泛用于 CNN、RNN 和 Transformer[176]。ALBERT[63]使用跨层参数共享和因式嵌入参数化来减少 PTM 的参数。虽然参数的数量大大减少，但 ALBERT 的训练和推理时间甚至比标准的 BERT 还要长。

通常，参数共享并不能提高推理阶段的计算效率。

4.5.4 知识提炼

知识提取（KD）[168]是一种压缩技术，其中一个称为学生模型的小模型被训练来重现一个称为教师模型的大模型的行为。在这里，教师模型可以是许多模型的集合，通常经过良好的预培训。与模型压缩不同，蒸馏技术通过一些优化目标从固定的教师模型学习一个小的学生模型，而压缩技术旨在搜索一个更稀疏的体系结构。通常，蒸馏机制可分为三种类型：（1）从软目标概率蒸馏，（2）从其他知识蒸馏，（3）从其他结构蒸馏：

（1） 从软目标概率中提取。Bucilua 等人[165]表明，让学生接近教师模型可以将知识从教师传递给学生。一种常用的方法是近似教

师模型的逻辑。DistilBERT[106]将蒸馏损失超过教师软目标概率的学生模型训练为：

$$\mathcal{L}_{\text{KD-CE}} = \sum_i t_i \cdot \log(s_i), \tag{5}$$

其中 ti 和 si 分别是教师模型和学生模型估计的概率。软目标概率的提取也可以用于特定于任务的模型，如信息检索[177]和序列标记[178]。

（2） 从其他知识中提炼。软目标概率的提取将教师模型视为一个黑匣子，只关注其输出。此外，分解教师模式和提炼更多知识可以改善学生模式。

Table 3: Comparison of Compressed PTMs

Method	Type	#Layer	Loss Function*	Speed Up	Params	Source PTM	GLUE‡
$\text{BERT}_{\text{BASE}}$ [16]	Baseline	12	$\mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{NSP}}$		110M		79.6
$\text{BERT}_{\text{LARGE}}$ [16]		24	$\mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{NSP}}$		340M		81.9
Q-BERT [104]	Quantization	12	HAWQ + GWQ	-		$\text{BERT}_{\text{BASE}}$	≈ 99% BERT°
Q8BERT [105]		12	DQ + QAT	-		$\text{BERT}_{\text{BASE}}$	≈ 99% BERT
ALBERT§ [63]	Param. Sharing	12	$\mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{SOP}}$	×5.6 ~ 0.3	12 ~ 235M		89.4 (ensemble)
DistilBERT [106]	Distillation	6	$\mathcal{L}_{\text{KD-CE}}$+$\text{Cos}_{\text{KD}}$+ $\mathcal{L}_{\text{MLM}}$	×1.63	66M	$\text{BERT}_{\text{BASE}}$	77.0 (dev)
TinyBERT§ † [107]		4	$\text{MSE}_{\text{embed}}$+$\text{MSE}_{\text{attn}}$+ MSE_{hidn}+$\mathcal{L}_{\text{KD-CE}}$	×9.4	14.5M	$\text{BERT}_{\text{BASE}}$	76.5
BERT-PKD [169]		3 ~ 6	$\mathcal{L}_{\text{KD-CE}}$+$\text{PT}_{\text{KD}}$+ $\mathcal{L}_{\text{Task}}$	×3.73 ~ 1.64	45.7 ~ 67 M	$\text{BERT}_{\text{BASE}}$	76.0 ~ 80.6♯
PD [170]		6	$\mathcal{L}_{\text{KD-CE}}$+$\mathcal{L}_{\text{Task}}$+ $\mathcal{L}_{\text{MLM}}$	×2.0	67.5M	$\text{BERT}_{\text{BASE}}$	81.2♯
MobileBERT§[171]		24	FMT+AT+PKT+ $\mathcal{L}_{\text{KD-CE}}$+$\mathcal{L}_{\text{MLM}}$	×4.0	25.3M	$\text{BERT}_{\text{LARGE}}$	79.7
MiniLM [108]		6	AT+AR	×1.99	66M	$\text{BERT}_{\text{BASE}}$	81.0♭
DualTrain§ †[172]		12	Dual Projection+$\mathcal{L}_{\text{MLM}}$	-	1.8 ~ 19.2M	$\text{BERT}_{\text{BASE}}$	75.8 ~ 81.9♯
BERT-of-Theseus [109]	Module Replacing	6	$\mathcal{L}_{\text{Task}}$	×1.94	66M	$\text{BERT}_{\text{BASE}}$	78.6

[1] The desing of this table is borrowed from [109, 173].
‡ The averaged score on 8 tasks (without WNLI) of GLUE benchmark (see Section 7.1). Here MNLI-m and MNLI-mm are regarded as two different tasks. 'dev' indicates the result is on dev set. 'ensemble' indicates the result is from the ensemble model.
* '$\mathcal{L}_{\text{MLM}}$ ', '$\mathcal{L}_{\text{NSP}}$', and '$\mathcal{L}_{\text{SOP}}$' indicate pre-training objective (see Section 3.1 and Table 1).'$\mathcal{L}_{\text{Task}}$' means task-specific loss.
'HAWQ', 'GWQ', 'DQ', and 'QAT' indicate Hessian AWare Quantization, Group-wise Quantization, Quantization-Aware Training, and Dynamically Quantized, respectively. 'KD' means knowledge distillation. 'FMT', 'AT', and 'PKT' mean Feature Map Transfer, Attention Transfer, and Progressive Knowledge Transfer, respectively. 'AR' means Self-Attention value relation.
§ The dimensionality of the hidden or embedding layers is reduced.
† Use a smaller vocabulary.
♭ Generally, the F1 score is usually used as the main metric of the QQP task. But MiniLM reports the accuracy, which is incomparable to other works.
° Result on MNLI and SST-2 only.
♯ Result on the other tasks except for STS-B and CoLA.
♮ Result on MRPC, MNLI, and SST-2 only.

TinyBERT[107]通过嵌入输出、隐藏状态和自我注意分布执行层到层蒸馏。MobileBERT[171]还利用软目标概率、隐藏状态和自我关注分布执行层到层蒸馏。MiniLM[108]从教师模型中提取自我注意分布和自我注意价值关系。

此外，其他模型通过多种方法提取知识。Sun 等人[169]引入了“患者”教师-学生机制，Liu 等人[179]利用 KD 改进了预先训练的多任务深层神经网络。

（3） 蒸馏到其他结构。一般来说，学生模型的结构与教师模型相同，只是层大小和隐藏大小较小。然而，不仅减少参数，而且将模型结构从 Transformer 简化为 RNN[180]或 CNN[181]可以降低计算复杂度。

4.5.5 模块更换

模块替换是一种有趣且简单的减小模型尺寸的方法，它用更紧凑的替代品替换原始 PTM 的大模块。Xu 等人[109]提出了 Theseus 压缩，其动机是一个著名的思想实验，名为“Theseus 之船”，该实验逐步用参数较少的模块替换源模型中的模块。与 KD 不同，Theseus 压缩只需要一个特定于任务的损失函数。Theseus 的 BERT 压缩模型在保持源模型 98%以上性能的同时，速度提高了 1.94 倍。

4.5.6 提前退出

另一种减少推理时间的有效方法是提前退出，它允许模型在出口匝道提前退出，而不是通过整个模型。要执行的层数取决于输入。

早期退出的思想首先应用于计算机视觉，如 BranchyNet[182]和浅-深网络[183]。随着深度预训练语言模型的出现，早期退出最近被用于加速基于转换器的模型。作为之前的工作，Universal Transformer[176]使用自适应计算时间（ACT）机制[184]实现输入自适应计算。Elbayad 等人[185]提出了用于机器翻译的深度自适应转换器，该转换器可以学习预测特定序列或标记需要多少解码层。Liu 等人[186]没有学习需要多少计算，而是提出了两种分别基于互信息（MI）和重建损失的估计方法，以直接将适当的计算分配给每个样本。

最近，DeeBERT[110]、RightTool[111]、FastBERT[112]、ELBERT[187]、PABEE[113]提出了减少变压器编码器计算量的建议语言理解任务。他们的方法通常包括两个步骤：（a）训练注入的出口（也称为内部分类器），和（b）设计退出策略以决定是否退出。

通常，训练目标是所有匝道交叉熵损失的加权和，即：

$$\mathcal{L}_{\text{early-exit}} = \sum_{i=1}^{M} w_i \cdot \mathcal{L}_i, \qquad (6)$$

式中，M 是出口匝道的数量。FastBERT[112]采用了自蒸馏损失，用最终分类器生成的软目标来训练每个出口匝道。廖等人[114]通过考虑过去和未来的信息，改进了目标。特别是，对出口匝道进行训练，以聚集过去层的隐藏状态，并近似未来层的隐藏状态。此外，Sun 等人[115]从集成学习和互信息的角度开发了一个新的训练目标，通过该目标，出口匝道被训练为一个集成。他们提出的目标不仅优化了每个出口匝道的准确性，还优化了出口匝道的多样性。

在推理过程中，需要一个退出策略来决定是提前退出还是继续到下一层。DeeBERT[110]，FastBERT[112]，Liao 等人[114]采用预测分布的熵作为现有标准。类似地，RightTool[111]使用最大 softmax 分数来决定是否退出。PABEE 开发了一种基于耐心的策略，当连续层的预测

不变时，允许样本退出。此外，Sun 等人[115]采用了一种基于投票的策略，让所有过去的出口进行投票，以决定是否退出。此外，Li 等人[116]提出了基于窗口的不确定性作为退出标准，以实现序列标记任务的令牌级提前退出（TokEE）。

5、 使 PTM 适应下游任务

尽管 PTM 从大量语料库中获取一般语言知识，但如何有效地使其知识适应下游任务仍然是一个关键问题。

5.1 迁移学习

迁移学习[188]是将源任务（或领域）中的知识调整到目标任务（或领域）中。图 4 给出了迁移学习的一个例子。

NLP 中的迁移学习有多种类型，如领域适应、跨语言学习、多任务学习。使 PTM 适应下游任务是顺序转移学习任务，其中任务按顺序学习，目标任务有标记数据。

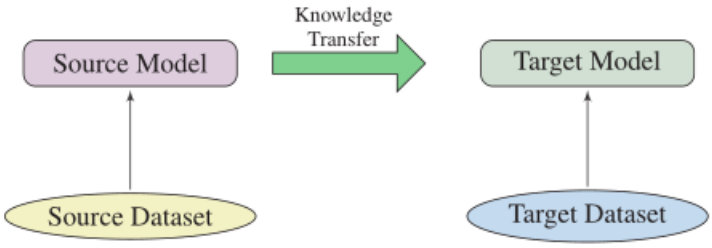

Figure 4: Transfer Learning

5.2 如何转移？

为了将 PTM 的知识转移到下游 NLP 任务，我们需要考虑以下问题：

5.2.1 选择合适的培训前任务、模型架构和语料库

不同的 PTM 通常对同一个下游任务有不同的影响，因为这些 PTM 使用不同的训练前任务、模型体系结构和语料库进行训练。

（1） 目前，语言模型是最流行的预训练任务，可以更有效地解决广泛的 NLP 问题[58]。然而，不同的训练前任务有自己的偏见，对不同的任务产生不同的影响。例如，NSP 任务[16]让 PTM 理解两句话之间的关系。因此，PTM 可以帮助下游任务，如问答（QA）和自然语言推理（NLI）。

（2） PTM 的体系结构对于下游任务也很重要。例如，虽然 BERT 有助于大多数自然语言理解任务，但很难生成语言。

（3） 下游任务的数据分布应近似于 PTMs。目前，有大量现成的 PTM，可以方便地用于各种特定领域或特定语言的下游任务。

因此，在给定目标任务的情况下，选择经过适当训练前任务、体系结构和语料库训练的 PTM 总是一个很好的解决方案。

5.2.2 选择合适的层

给定一个预先训练好的深度模型，不同的层应该捕获不同类型的信息，例如词性标记、解析、长期依赖、语义角色、相互引用。对于基于 RNN 的模型，Belinkov et al.[189]和 Melamud et al.[34]表明，在多层 LSTM 编码器中从不同层学习的表示有利于不同的任务（例如，预测词性标签和理解词义）。对于基于变压器的 PTM，Tenney 等人[122]发现 BERT 代表了传统 NLP 管道的步骤：句法信息在网络中出现较早，而高级语义信息出现在更高层。

设 H（l）（16l6l）表示具有 l 层的预训练模型的第 l 层表示，g（·）表示目标任务的特定任务模型。选择表示有三种方法：a）仅嵌入。一种方法是只选择预先训练好的静态嵌入，而模型的其余部分仍然需要从头开始为新的目标任务进行训练。它们无法捕获可能更有用的更高级别的信息。单词嵌入只在捕捉单词的语义时有用，但我们还需要理解更高层次的概念，比如词义。b） 顶层。最简单有效的方法是将顶层的表示输入到特定于任务的模型 g（H（L））中。c） 所有层。一种更灵活的方法是在软版本中自动选择最佳层，如 ELMo[14]：

$$\mathbf{r}_t = \gamma \sum_{l=1}^{L} \alpha_l \mathbf{h}_t^{(l)}, \qquad (7)$$

其中，αl 是层 l 的 softmax 归一化权重，γ 是一个标量，用于缩放预训练模型输出的向量。混合表示被输入到特定于任务的模型 g（rt）中。

5.2.3 调整还是不调整？

目前，有两种常见的模型转换方法：特征提取（预训练参数被冻结）和微调（预训练参数被解冻和微调）。在特征提取方面，将预先训练好的模型视为现成的特征抽取器。此外，暴露内部层很重要，因为它们通常编码最可转移的表示[190]。虽然这两种方法都可以显著地使大多数 NLP 任务受益，但特征提取方法需要更复杂的特定于任务的体系结构。因此，对于许多不同的下游任务，微调方法通常比特征提取方法更通用、更方便。

表 4 给出了适应 PTM 的一些常见组合。

Table 4: Some common combinations of adapting PTMs.

Where	FT/FE?†	PTMs
Embedding Only	FT/FE	Word2vec [11], GloVe [12]
Top Layer	FT	BERT [16], RoBERTa [43]
Top Layer	FE	BERT§ [191, 192]
All Layers	FE	ELMo [14]

† FT and FE mean Fine-tuning and Feature Extraction respectively.
§ BERT used as feature extractor.

5.3 随 PT 深度增加的微调策略

随着 PTM 深度的增加，它们捕获的表示使下游任务变得更容易。因此，整个模型的任务特定层很简单。自 ULMFit 和 BERT 以来，微调已成为 PTMs 的主要自适应方法。然而，微调过程往往很脆弱：即使具有相同的超参数值，不同的随机种子也可能导致截然不同的结果[193]。除了标准的微调，还有一些有用的微调策略。

两阶段微调另一种解决方案是两阶段转移，它在预培训和微调之间引入一个中间阶段。在第一阶段，PTM 被转换成一个由中间任务或语料库微调的模型。第二阶段，根据目标任务对转移模型进行微调。Sun 等人[64]表明，对相关领域语料库的“进一步预训练”可以进一步提高 BERT 的能力，并在八个广泛研究的文本分类数据集上实现最先进的性能。Phang 等人[194]和 Garg 等人[195]引入了与目标任务相关的中间监督任务，这给 BERT、GPT 和 ELMo 带来了很大的改进。Li 等人[65]也使用了两阶段转移来预测故事的结局。所提出的 TransBERT（transferable-BERT）不仅可以从大规模未标记数据中转移一般语言知识，还可以从各种语义相关的监督任务中转移特定类型的知识。

多任务微调 Liu 等人[67]在多任务学习框架下微调了 BERT，这表明多任务学习和预训练是互补的技术。

使用额外的自适应模块进行微调微调的主要缺点是参数效率低下：每个下游任务都有自己的微调参数。因此，更好的解决方案是在原始参数固定的情况下，向 PTM 中注入一些可微调的自适应模块。

Stickland 和 Murray[68]为一个单一共享的 BERT 模型配备了额外的小型任务特定适应模块，即投射注意力层（PAL）。与 PAL 共享的 BERT 与 GLUE 基准上单独微调的模型相匹配，参数大约减少了 7 倍。类似地，Houlsby 等人[69]通过添加适配器模块修改了预训练的 BERT 的体系结构。适配器模块产生一个紧凑且可扩展的模型；它们只为每个任务添加了几个可训练的参数，并且可以添加新任务，而无需重新访问以前的任务。原始网络的参数保持不变，从而产生高度的参数共享。

其他人受到广泛使用的集成模型成功的推动，Xu 等人[196]通过两种有效机制改进了 BERT 的微调：自集成和自蒸馏，这可以在不利用外部资源或显著降低训练效率的情况下提高 BERT 在下游任务上的性能。他们在一个训练过程中整合了合奏和提炼。教师模型是一个集合模型，通过对之前时间步长中的几个学生模型进行参数平均。

逐步解冻[38]不是同时微调所有层，而是从顶层开始逐步解冻PTM层的有效方法。Chronopoulou等人[197]提出了一种更简单的解冻方法，即顺序解冻，该方法首先微调随机初始化的任务特定层，然后解冻PTM的隐藏层，最后解冻嵌入层。

Li和Eisner[198]使用可变信息瓶颈压缩ELMo嵌入，同时只保留有助于目标任务的信息。总体而言，上述工作表明，更好的微调策略可以进一步激发PTM的效用。

5.3.1 基于提示的调整

缩小预培训和微调之间的差距可以进一步提高PTM在下游任务中的性能。另一种方法是通过设计适当的提示，将下游任务重新格式化为传销任务。基于提示的方法在少数镜头设置[199,200,70,72]、零镜头设置[129,201]甚至完全监督设置[74,75]中表现出了强大的功能。根据提示是离散的还是连续的，当前基于提示的方法可以分为两个分支。

离散提示离散提示是插入到输入文本中的一系列单词，这有助于PTM更好地模拟下游任务。Sun等人[202]通过将基于方面的情绪分析（ABSA）任务转换为句子对分类任务，构建了一个辅助句子，但其模型参数仍需微调。GPT-3[59]提出了上下文学习，将原始输入与任务描述和几个示例连接起来。通过这种方式，GPT-3可以在不调整参数的情况下获得具有竞争力的性能。此外，Petroni等人[129]发现，在适当的手动提示下，BERT可以在没有训练的情况下很好地完成实体预测任务（LAMA）。除了LAMA，Schick和Sch¨utze[200,70]还提出了PET，为各种文本分类和蕴涵任务设计了离散提示。然而，手动设计的提示可能不太理想，因此，开发了许多方法来自动生成提示。LPAQA[201]使用两种方法，即基于挖掘的生成和基于释义的生成，来找到表达特定关系的最佳模式。AutoPrompt[71]通过梯度引导搜索找到最佳提示。LM-BFF[72]使用T5[42]自动生成提示。

连续提示不是寻找最佳的具体提示，另一种选择是在连续空间中直接优化提示，即提示向量不一定是PTM的字型嵌入。优化后的连续提示与字型嵌入连接在一起，然后输入到PTM中。秦和艾斯纳[203]和钟等[204]发现，优化后的连续提示在关系任务上的表现优于具体提示（包括手动[129]、挖掘提示（LPAQA[201]）和梯度搜索提示（自动提示[71]）。W ARP[73]在输入序列之前、之间和之后插入可训练的连续提示令牌，同时保持PTM的参数不变，从而在GLUE基准测试中获得可观的性能。

前缀调整[74]插入连续提示作为 GPT-2 输入的前缀，用于表到文本生成和 BART 的摘要。前缀调整作为一种高效的参数调整技术，在全监督设置下取得了相当的性能，在少镜头设置下优于模型微调。此外，P-Tuning[75]表明，在连续提示下，GPT 在自然语言理解（NLU）任务中也可以达到与类似大小的 BERT 相当甚至更好的性能。最近，Lester 等人[205]表明，快速调谐与规模相比更具竞争力。当 PTM 超过数十亿个参数时，模型微调和提示微调之间的差距可以缩小，这使得基于提示的微调成为高效服务于大规模 PTM 的一种非常有前途的方法。

6、 PTMs 的资源

网上有很多关于 PTM 的相关资源。表 5 提供了一些流行的存储库，包括第三方实现、论文列表、可视化工具以及 PTMs 的其他相关资源。

此外，还有其他一些关于 NLP 的 PTM 的优秀调查论文[211、212、173]。

7、 软件

在本节中，我们总结了 PTMs 在几个经典 NLP 任务中的一些应用。

Table 5: Resources of PTMs

Resource	Description	URL
	Open-Source Implementations [§]	
word2vec	CBOW,Skip-Gram	https://github.com/tmikolov/word2vec
GloVe	Pre-trained word vectors	https://nlp.stanford.edu/projects/glove
FastText	Pre-trained word vectors	https://github.com/facebookresearch/fastText
Transformers	Framework: PyTorch&TF, PTMs: BERT, GPT-2, RoBERTa, XLNet, etc.	https://github.com/huggingface/transformers
Fairseq	Framework: PyTorch, PTMs:English LM, German LM, RoBERTa, etc.	https://github.com/pytorch/fairseq
Flair	Framework: PyTorch, PTMs:BERT, ELMo, GPT, RoBERTa, XLNet, etc.	https://github.com/flairNLP/flair
AllenNLP [206]	Framework: PyTorch, PTMs: ELMo, BERT, GPT-2, etc.	https://github.com/allenai/allennlp
fastNLP	Framework: PyTorch, PTMs: RoBERTa, GPT, etc.	https://github.com/fastnlp/fastNLP
UniLMs	Framework: PyTorch, PTMs: UniLM v1&v2, MiniLM, LayoutLM, etc.	https://github.com/microsoft/unilm
Chinese-BERT [85]	Framework: PyTorch&TF, PTMs: BERT, RoBERTa, etc. (for Chinese)	https://github.com/ymcui/Chinese-BERT-wwm
BERT [16]	Framework: TF, PTMs: BERT, BERT-wwm	https://github.com/google-research/bert
RoBERTa [43]	Framework: PyTorch	https://github.com/pytorch/fairseq/tree/master/examples/roberta
XLNet [49]	Framework: TF	https://github.com/zihangdai/xlnet/
ALBERT [63]	Framework: TF	https://github.com/google-research/ALBERT
T5 [42]	Framework: TF	https://github.com/google-research/text-to-text-transfer-transformer
ERNIE(Baidu) [84, 153]	Framework: PaddlePaddle	https://github.com/PaddlePaddle/ERNIE
CTRL [207]	Conditional Transformer Language Model for Controllable Generation.	https://github.com/salesforce/ctrl
BertViz [208]	Visualization Tool	https://github.com/jessevig/bertviz
exBERT [209]	Visualization Tool	https://github.com/bhoov/exbert
TextBrewer [210]	PyTorch-based toolkit for distillation of NLP models.	https://github.com/airaria/TextBrewer
DeepPavlov	Conversational AI Library. PTMs for the Russian, Polish, Bulgarian, Czech, and informal English.	https://github.com/deepmipt/DeepPavlov
	Corpora	
OpenWebText	Open clone of OpenAI's unreleased WebText dataset.	https://github.com/jcpeterson/openwebtext
Common Crawl	A very large collection of text.	http://commoncrawl.org/
WikiEn	English Wikipedia dumps.	https://dumps.wikimedia.org/enwiki/
	Other Resources	
Paper List		https://github.com/thunlp/PLMpapers
Paper List		https://github.com/tomohideshibata/BERT-related-papers
Paper List		https://github.com/cedrickchee/awesome-bert-nlp
Bert Lang Street	A collection of BERT models with reported performances on different datasets, tasks and languages.	https://bertlang.unibocconi.it/

[§] Most papers for PTMs release their links of official version. Here we list some popular third-party and official implementations.

7.1 一般评价基准

NLP 社区面临的一个重要问题是，我们如何以可比的标准评估 PTM。因此，大规模基准测试是必要的。通用语言理解评估（GLUE）基准[213]是九项自然语言理解任务的集合，包括单句分类任务（CoLA 和 SST-2）、成对文本分类任务（MNLI、RTE、WNLI、QQP 和 MRPC）、文本相似性任务（STSB）和相关排名任务（QNLI）。GLUE benchmark 经过精心设计，用于评估模型的稳健性和泛化性。GLUE 不为测试集提供标签，而是设置一个评估服务器。然而，由于近年来的进步极大地削弱了 GLUE 基准的净空，一个名为 SuperGLUE[214]的新基准被提出。与 GLUE 相比，SuperGLUE 具有更具挑战性的任务和更多样化的任务格式（例如，共指消解和问答）。最先进的 PTM 列于相应的排行榜 4）5）。

7.2 问答

问答（QA）或狭义的概念机器阅读理解（MRC）是 NLP 社区中的一个重要应用。从简单到困难，QA 任务有三种类型：单轮抽取式 QA（团队）【215】、多轮生成式 QA（CoQA）【216】和多跳 QA（HotpotQA）【217】。

BERT 创造性地将提取 QA 任务转换为预测答案起始跨度和结束跨度的跨度预测任务[16]。在那之后，PTM 作为预测跨度的编码器已经成为一个有竞争力的基线。对于抽取式 QA，Zhang 等人[218]提出了透视读卡器架构，并用 PTM（如 ALBERT）初始化编码器。对于多轮生成性 QA，Ju 等人[219]提出了“PTM+对抗性训练+理由标记+知识提炼”模型。对于多跳 QA，Tu 等人[220]提出了一种可解释的“选择、回答和解释”（SAE）系统，PTM 充当选择模块中的编码器。通常，在所提出的 QA 模型中，编码器参数通过 PTM 初始化，其他参数则随机初始化。最先进的模型列在相应的排行榜上。6) 7) 8)

7.3 情绪分析

通过简单地微调 SST-2，BERT 的表现优于之前的最先进模型，SST-2 是情绪分析（SA）中广泛使用的数据集[16]。Bataa 和 Wu[221]利用 BERT 和转移学习技术，实现了日本 SA 的新水平。

尽管他们在简单情绪分类方面取得了成功，但直接将 BERT 应用于基于方面的情绪分析（ABSA），这是一项细粒度的 SA 任务，显示出不太显著的改进[202]。为了更好地利用 BERT 的强大表示，Sun 等人[202]通过将 ABSA 从单句分类任务转换为句子对分类任务，构造了

一个辅助句。Xu 等人[222]提出了岗位培训，以使 BERT 从其源域和任务适应 ABSA 域和任务。此外，Rietzler 等人[223]扩展了[222]的工作，用 ABSA 性能分析了跨域岗位培训的行为。Karimi 等人[224]表明，训练后的 BERT 的表现可以通过对抗性训练进一步提高。Song 等人[225]添加了一个额外的池模块，可以作为 LSTM 或注意机制来实现，以利用 ABSA 的中间层。此外，Li 等人[226]联合学习了面向端到端 ABSA 的方面检测和情感分类。SentiLR[79]从 SentiWordNet 获取词性标签和先验情感极性，并采用标签感知传销，利用引入的语言知识捕捉句子级情感标签和单词级情感转移之间的关系。SentiLR 在几个句子和体层情绪分类任务上实现了最先进的性能。

对于情绪传递，Wu 等人[227]在 BERT 的基础上提出了“伪装和填充”。在屏蔽步骤中，该模型通过屏蔽情感标记将情感与内容分离。在填充步骤中，它使用 BERT 和目标情绪嵌入来填充蒙面位置。

7.4 命名实体识别

命名实体识别（NER）是信息抽取中的一个重要组成部分，在许多 NLP 下游任务中发挥着重要作用。在深度学习中，大多数 NER 方法都在 sequencelabeling 框架中。将句子中的实体信息转换为标签序列，一个标签对应一个单词。该模型用于预测每个单词的标签。由于 ELMo 和 BERT 已经在 NLP 中展示了他们的力量，所以有很多关于 NER 的预训练模型的工作。

Akbik 等人[37]使用预先训练的字符级语言模型为 NER 生成单词级嵌入。TagLM[228]和 ELMo[14]使用预先训练好的语言模型的最后一层输出和每层输出的加权和作为单词嵌入的一部分。Liu 等人[229]使用分层修剪和密集连接来加速 ELMo 对 NER 的推断。Devlin 等人[16]使用第一个 BPE 的 BERT 表示来预测没有 CRF 的每个单词的标签。Pires 等人[150]通过多语言学习实现了零射击。Tsai 等人[178]利用知识蒸馏在单个 CPU 上运行一个小型的 BERT for NER。此外，BERT 还用于特定领域的 NER，如生物医学[230100]等。

7.5 机器翻译

机器翻译（MT）是 NLP 领域的一项重要任务，吸引了许多研究人员。几乎所有的神经机器翻译（NMT）模型共享编码器解码器框架，其首先将编码器的输入令牌编码为隐藏的表示，然后从解码器中解码目标语言中的输出令牌。Ramachandran 等人[36]发现，通过使用

预先训练好的两种语言模型的权重初始化编码器和解码器，可以显著改进编码器-解码器模型。Edunov 等人[231]使用 ELMo 在 NMT 模型中设置单词嵌入层。本文通过使用预先训练好的语言模型进行源词嵌入初始化，展示了英语-土耳其语和英语-德语 NMT 模型的性能改进。
鉴于 BERT 在其他 NLP 任务中的出色性能，研究如何将 BERT 纳入 NMT 模型是很自然的。Conneau 和 Lample[46]试图通过一个多语言预训练的 BERT 模型初始化整个编码器和解码器，并表明在无监督机器翻译和英罗监督机器翻译上可以实现显著的改进。同样，Clinchant 等人[232]设计了一系列通过不同的实验来检验在 NMT 模型的编码器部分使用 BERT 的最佳策略。通过使用 BERT 作为编码器的初始化，他们实现了一些改进。此外，他们发现这些模型在域外数据集上可以获得更好的性能。Imamura 和 Sumita[233]提出了 NMT 的两阶段 BERT 微调方法。在第一阶段，编码器由预先训练好的伯特模型初始化，它们只在训练集上训练解码器。在第二阶段，在训练集上对整个 NMT 模型进行联合微调。实验表明，该方法优于直接微调整个模型的单级微调方法。除此之外，Zhu 等人[192]建议使用预先训练好的 BERT 作为额外记忆，以促进 NMT 模型的建立。具体来说，他们首先用预先训练好的 BERT 对输入标记进行编码，并将最后一层的输出用作额外内存。然后，NMT 模型可以通过编码器和解码器的每一层中的额外注意模块访问内存。它们在有监督、半监督和无监督机器翻译方面都有显著的改进。

MASS（掩蔽序列到序列预训练）[41]利用 Seq2Seq MLM 联合预训练编码器和解码器，而不是仅预训练编码器。在实验中，这种方法在无监督机器翻译和英语-罗马尼亚语监督机器翻译上都可以超过 Conneau 和 Lample[46]提出的伯特式预训练。与 MASS 不同，BART[50]的多语言扩展 mBART[61]，在 25 种语言的大规模单语语料库上，与 Seq2Seq 去噪自动编码器（DAE）任务一起预训练编码器和解码器。实验表明，mBART 能够在句子和文档两个层面上显著提高有监督和无监督机器翻译。

7.6 总结

摘要，旨在产生一个较短的文本，从而保留较长文本的最大意义，近年来引起了 NLP 社区的关注。自从 PTM 广泛使用以来，这项任务得到了显著改善。Zhong 等人[191]引入可转移知识（如 BERT）进行总结，并超越了以前的模型。Zhang 等人[234]尝试预先训练一个文档级模型，预测句子而不是单词，然后将其应用于下游任务，如摘要。更详细地说，

Zhang 等人[163]为预训练设计了一个间隙句生成（GSG）任务，其目标是从输入中生成类似摘要的文本。此外，刘和拉帕塔[235]提出了贝尔特森。BERTSUM 包括一个新的文档级编码器，以及一个用于提取摘要和抽象摘要的通用框架。在编码器帧中，BERTSUM 通过插入多个[CLS]标记来学习句子表示来扩展 BERT。为了提取摘要，BERTSUM 将几个句子间转换层堆叠起来。对于抽象总结，BERTSUM 提出了一种两阶段微调方法，使用一种新的微调时间表。Zhong 等人[236]提出了一个新的摘要级框架 MA TCHSUM，并将抽取摘要概念化为一个语义文本匹配问题。他们提出了一个暹罗 BERT 体系结构来计算源文档和候选摘要之间的相似性，并通过仅使用 BERT 的基本版本，在 CNN/DailyMail（ROUGE-1 中为 44.41）上获得了最先进的结果。

7.7 对抗性攻击和防御

深层神经模型容易受到敌对示例的影响，这些示例可能会误导模型产生特定的错误预测，并对原始输入产生难以察觉的干扰。在 CV 中，对抗性攻击和防御被广泛研究。然而，由于语言的离散性，这对文本来说仍然是一个挑战。为文本生成对抗性样本需要具备以下品质：（1）人类法官无法察觉，但对神经模型具有误导性；（2） 语法流利，语义与原始输入一致。Jin 等人[237]用对抗性的例子成功地攻击了文本分类和文本蕴涵方面的微调 BERT。Wallace 等人[238]定义了通用的对抗性触发器，当连接到任何输入时，该触发器可以诱导模型产生特定目的的预测。一些触发器甚至会导致 GPT-2 模型生成种族主义文本。Sun 等人[239]表明，BERT 对拼写错误并不敏感。

PTM 还具有生成对抗性样本的巨大潜力。Li 等人[240]提出了 BERT 攻击，这是一种基于 BERT 的高质量有效攻击者。他们在下游任务中将 BERT 与另一个经过微调的 BERT 进行对比，并成功误导目标模型进行错误预测，在成功率和干扰百分比方面均优于最先进的攻击策略，同时生成的对抗性样本流畅且语义保留。

此外，PTM 的对抗性防御也很有前景，它提高了 PTM 的鲁棒性，使其对对抗性攻击具有免疫力。

对抗性训练旨在通过最小化嵌入空间中标签保留扰动的最大风险来提高泛化能力。最近的研究[241，242]表明，对抗性预训练或微调可以提高自然语言处理 PTM 的泛化和鲁棒性。

8、 未来方向

尽管 PTM 已经证明了其对各种 NLP 任务的能力，但由于语言的复杂性，挑战仍然存在。在本节中，我们提出了 PTM 的五个未来方向。

（1）目前，PTM 尚未达到上限。通过更多的培训步骤和更大的语料库，可以进一步改进大多数当前的 PTM。

通过增加模型的深度，可以进一步提升 NLP 的技术水平，例如威震天 LM[243]（83 亿个参数，72 个变压器层，隐藏大小为 3072 和 32 个注意头）和图灵 NLG9（170 亿个参数，78 个变压器层，隐藏大小为 4256 和 28 个注意头）。通用 PTM 一直是我们学习语言固有的普遍知识（甚至世界知识）的追求。然而，此类 PTM 通常需要更深入的体系结构、更大的语料库和具有挑战性的训练前任务，这进一步导致更高的训练成本。然而，训练大型模型也是一个具有挑战性的问题，需要更复杂和高效的训练技术，如分布式训练、混合精度、梯度累积等。因此，更实用的方向是设计更高效的模型体系结构、自我监督的预训练任务、优化器，以及使用现有硬件和软件的训练技能。ELECTRA[56]是朝着这个方向发展的一个很好的解决方案。

（2） PTMs 体系结构变压器已被证明是一种有效的预培训体系结构。然而，变压器的主要限制是其计算复杂度，它与输入长度成二次关系。由于 GPU 内存的限制，目前大多数 PTM 无法处理超过 512 个令牌的序列。打破这一限制需要改进变压器的结构。尽管许多工作[25]试图提高变压器的效率，但仍有很大的改进空间。

此外，为 PTM 寻找更有效的替代非变压器体系结构对于获取更长范围的上下文信息非常重要。深层体系结构的设计具有挑战性，我们可能会寻求一些自动方法的帮助，例如神经体系结构搜索（NAS）[245]。

（3） 面向任务的预训练和模型压缩在实践中，不同的下游任务需要不同的 PTMs 能力。PTMs 和下游任务之间的差异通常存在于两个方面：模型体系结构和数据分布。更大的差异可能导致 PTM 的益处可能微不足道。例如，文本生成通常需要一个特定的任务来预训练编码器和解码器，而文本匹配则需要为句子对设计的预训练任务。

此外，虽然较大的 PTM 通常可以带来更好的性能，但一个实际问题是如何在特殊情况下利用这些巨大的 PTM，例如低容量设备和低延迟应用程序。因此，我们可以为下游任务仔细设计特定的模型架构和预训练任务，或者从现有的 PTM 中提取部分特定于任务的知识。

我们不需要从头开始培训面向任务的 PTM，而是可以通过使用模型压缩等技术（见第 4.5 节）使用现有的通用 PTM 来教授他们。尽管 CV[246]

中广泛研究了 CNN 的模型压缩，但 NLP 的 PTM 压缩才刚刚开始。变压器的全连接结构也使模型压缩更具挑战性。

（4） 知识转移超出微调目前，微调是将 PTM 的知识转移到下游任务的主要方法，但其不足之处在于参数效率低下：每个下游任务都有自己的微调参数。一种改进的解决方案是固定 PTM 的原始参数，并为特定任务添加小型可微调自适应模块[68,69]。因此，我们可以使用共享的 PTM 来服务多个下游任务。事实上，从 PTM 中挖掘知识可以更加灵活，例如特征提取、知识提炼[210]，数据扩充[247，248]，使用 PTM 作为外部知识[129]。预计会有更有效的方法。

（5）PTM 的可解释性和可靠性虽然 PTM 达到了令人印象深刻的性能，但其深刻的非线性结构使得决策过程高度不透明。

最近，可解释人工智能（XAI）[249]已成为一般人工智能领域的一个热点。与用于图像的 CNN 不同，由于变压器式结构和语言的复杂性，解读 PTM 更难。已经做出了大量努力（见第 3.3 节）来分析 PTMs 中包含的语言和世界知识，这有助于我们在一定程度上透明地理解这些 PMT。然而，关于模型分析的许多工作都依赖于注意机制，注意对解释性的有效性仍然存在争议[250251]。

此外，PTM 也容易受到敌对攻击（见第 7.7 节）。随着 PTM 在生产系统中的广泛使用，PTM 的可靠性也成为一个备受关注的问题。对 PTM 的对抗性攻击的研究有助于我们充分了解它们的能力他们的弱点。PTM 的对抗性防御也很有前景，它提高了 PTM 的健壮性，使其对对抗性攻击免疫。

总的来说，作为许多 NLP 应用中的关键组件，PTM 的可解释性和可靠性在许多方面有待进一步探索，这有助于我们了解 PTM 的工作原理，并为更好地使用和进一步改进提供指导。

9、 结论

在本次调查中，我们对 NLP 的 PTM 进行了全面概述，包括背景知识、模型体系结构、培训前任务、各种扩展、适应方法、相关资源和应用。基于当前的 PTM，我们从四个不同的角度提出了一种新的 PTM 分类法。我们还提出了 PTMs 未来可能的研究方向。