arXiv:2302.08575v1 [cs.CL] 16 Feb 2023

Gerhard Paaß 和 Sven Giesselbach

自然语言处理基础模型

融合媒体的预训练语言模型

本书已获施普林格·自然接受，并将作为开放获取专著出版。
https://link.springer.com/book/9783031231896。本书根据知识共享署名-非商业性使用-相同方式共享许可协议（
https://creativecommons.org/licenses/by-nc-sa/4.0/）授权，但其中收录的其他作者的材料除外，这些材料可能适用不同的许可协议。

施普林格·自然

前言

多年来，人工智能（“AI”），尤其是机器学习，一直是科学界、商业界和社会共同关注的焦点；对许多人而言，它们或许像一位我们逐渐了解并欣赏其能力的老朋友。毕竟，如今基于机器学习的人工智能似乎几乎无处不在。当我们浏览社交媒体上的时间线、听音乐或观看电影时，机器学习算法会为我们提供推荐。我们通过移动电话上的数字助理交谈时，它们能够将我们的语音转录成文字，并回答简单的问题。在某些情况下，人工智能系统有时能做出比人类医生更好的诊断；而在幕后，它们还驱动着如今商业管理、生产和物流领域的许多数字系统。也许，我们中的一些人甚至已经在最新的机动车中使用由机器学习驱动的半自动驾驶能力。

这些应用令人印象深刻——然而，另一场革命已经在路上。人工智能技术的新新浪潮即将彻底改变我们对人工智能系统能力的认识：基础模型。到目前为止，人工智能系统通常通过在专门为某项具体任务构建的数据集上训练学习算法来开发；而如今，研究人员和工程师正利用互联网上几乎无限量的可用数据、文档和图像，训练出相对独立于其未来可能用途的模型。这些深度网络模型使用包含数万亿个词语的大型文档集合，并结合数千亿个参数，构建对其输入的表示，并以一种之后能够将其用于不同任务的方式存储这些表示，例如问答，甚至推理。此类模型已经能够产生过去难以想象的结果，并将促成更加灵活、功能强大得多、最终也更接近真正通用人工智能的人工智能系统。

本书对基础模型主题进行了出色而深入的介绍，详细阐述了此类模型的主要类别，以及它们在文本、语音、图像和视频方面的应用。因此，对于有兴趣进入这一领域的读者，本书可作为概览；对于希望进一步了解各种具体方法的读者，也可作为更为详尽的参考。愿本书有助于

让基础模型为更广泛的读者所获取，从而帮助进一步传播和发展这项令人振奋的科技！

波恩，2022年7月 *Stefan Wrobel*教授

前言

四十年前，深度神经网络问世时，人们希望将其作为一种通用计算设备，用来模拟大脑的运作方式。然而，由于当时计算机的能力有限，它们只能用于解决小型问题，随后便淡出了科学研究的视野。

直到大约十年前，一种变体——卷积神经网络——才成功地在图像物体识别方面超越了其他方法。这得益于大型人工标注图像训练集的问世、图形处理器强大的计算能力，以及新型优化技术的高效性。此后不久，许多专用模型在其他领域提升了性能，例如用于预测序列的循环神经网络，以及用于操控视频游戏的强化学习模型。然而，这些深度神经网络在大多数情况下的结果并不理想，通常无法达到人类表现的水平。

语言处理领域尤其受益于这样一种理念：用一个长向量，即嵌入，来表示每个单词的意义。五年前，Google 的工程师对这种方法进行了重大改进。他们将这些嵌入与其他词语的嵌入相关联，从而能够在下一层计算出新的嵌入，使某个单词的嵌入能够根据上下文进行调整。例如，“bank”在“money”附近通常指金融机构，而在“river”附近则指倾斜的河岸。这种操作称为自注意力，使模型能够获取前所未有的大量语义信息。模型不再逐词处理文本，而是一次性关联所有词语，从而提高了处理速度。

这些模型可以用作语言模型，根据文本中前面的词语预测下一个词。它们不需要人工标注，可以在纯文本上进行训练，例如来自互联网的文本。事实证明，这些模型规模越大、处理的训练文本越多，性能就越好。GPT-3 模型是一个里程碑，它拥有 1750 亿个参数，并使用 570 GB 的文本进行训练。它能够生成在句法和语义上都令人信服的文本，几乎无法与人工生成的文本区分开来。

进一步的实验表明，除了文本之外，这些模型还可以应用于其他类型的序列，例如图片、视频、声音录音或分子序列。每次，模型都会将小型输入分块表示为嵌入，并通过自注意力学习这些图像块之间的关系。由于这一过程可以同时针对不同媒体进行，嵌入便充当了共同的跨媒体表示。早期的深度神经网络通常是为单一任务设计的，而这些模型可以应用于各种任务，因此通常被称为“基础模型”。它们有望通过单一的高性能模型捕捉文本、语音、图像以及环境的感官印象，从而接近神经网络最初的愿景。

本书旨在介绍基于大量训练数据预训练的语言模型。如果这些模型的参数数量足够多，就称为基础模型。基础模型只需通过说明即可执行新任务，而且还能处理不同的媒体类型。本书尤其介绍了技术词汇，同时也涵盖概念、方法和网络架构。此外，本书还介绍了改进模型的方法，并探讨模型的性能及其弱点。书中有一大章节概述了基础模型在各种语言处理任务中的应用。最后，本书介绍了基础模型在跨媒体处理方面的能力。

本书帮助熟悉文本和媒体处理基础知识的研究人员和决策者参与语言模型和基础模型的设计，并更好地评估模型性质及其影响。对于数据分析师、学生、工程师和研究人员而言，本书是了解更深入专业文献的理想入门读物。

致谢

本书的出版离不开弗劳恩霍夫智能分析与信息系统研究所 IAIS（位于圣奥古斯丁）所营造的积极进取、充满专业激励的环境。我们要感谢所有支持本书项目的同事，以及来自个人生活圈的朋友们，无论是通过专业讨论、校对个别章节，还是提出宝贵意见：Katharina Beckh、Ewald Bindereif、Eduardo Brito、Nilesh Chakraborty、Heike Horstmann、Birgit Kirsch、Katrin Klug 和 Najmeh Mousavi。特别感谢 Heike Horstmann，她为本书的结构提供了宝贵建议，并克服诸多行政困难，组织了本书的开源出版。

本研究由德国联邦教育与研究部资助，是机器学习能力中心 ML2R（01IS18038B）项目的一部分。这项慷慨的支持让我们有时间深入研究基础模型。与研究中心同事们富有启发性的讨论，让我们关注到这一主题的许多方面。

但最大的感谢要献给我们的家人，是他们在漫长的写作时间里给予了我们必要的空间。特别是我，Gerhard Paaß，想感谢我的妻子

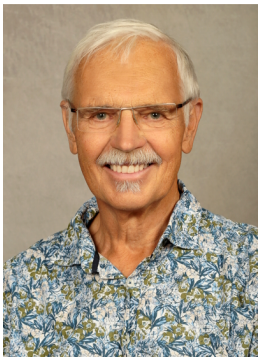
Margret Paaß，她的耐心与鼓励在本书的成功中发挥了重要作用，并且从规划阶段到校样校对，一直给予我们不可或缺的帮助。没有你的鼓励和支撑，我们不可能完成这本书。非常感谢你一直以来的支撑！

圣奥古斯丁，
2022年7月

Gerhard Paaß
Sven Giesselbach

作者简介

Gerhard Paaß博士 毕业于波恩大学数学与计算机科学专业，并撰写了关于经济模型预测准确性的博士论文。此后，他加入了数学与数据处理协会（GMD），该机构现为位于圣奥古斯丁的弗劳恩霍夫智能分析与信息系统研究所 IAIS。他曾担任多个研究项目的项目负责人，研究主题包括不确定性推理、多媒体神经网络和预测不确定性，并创立了 IAIS 的文本挖掘小组。Paaß 博士曾在中国、美国、澳大利亚和日本的大学进行多次访问研究。他发表了大量论文，并在人工智能领域获得过多个最佳论文奖。此外，多年来他一直担任讲师，并在弗劳恩霍夫大数据与人工智能联盟框架内，对明确数据科学家的全新职业描述发挥了非常重要的作用，也成功推动其在德国落地。他最近用德语撰写了一本关于“人工智能”的书，该书即将以英语出版。作为弗劳恩霍夫 IAIS 的首席科学家，Paaß 博士为该领域众多课程体系的开发作出了贡献。



Winfried Schneider，Fotostudio S2，波恩

SvenGiesselbach 是弗劳恩霍夫智能分析与信息系统研究所（IAIS）自然语言理解（NLU）团队的负责人，专攻人工智能和自然语言处理。他的团队致力于开发医疗、法律风格和通用文档理解领域的解决方案。Sven Giesselbach 也是莱茵—鲁尔机器学习能力中心（ML2R）的一员，在那里担任研究科学家，研究知识引导机器学习这一范式，并探索如何将知识注入机器学习模型，同时结合语言建模。他曾

发表了 10 多篇关于自然语言处理和理解的论文，聚焦于创作可用于实际应用的自然语言理解系统，以及如何在解决方案设计的不同阶段整合专家知识。Sven Giesselbach 曾领导自然语言理解展示平台的开发，这是一个用于展示最先进水平的自然语言理解模型的平台。他经常在暑期学校、会议和人工智能聚会上发表关于自然语言理解的演讲。



目录

1 引言 . 1

1.1 本书范围.1 1.2 文本预处理.4 1.3 向量空间模型与文档分类.5 1.4 非线性分类器.7 1.5 生成静态词嵌入.8 1.6 循环神经网络.10 1.7 卷积神经网络.12 1.8 总结.13

2 预训练语言模型 .19

2.1 BERT：自注意力与上下文嵌入.20 2.1.1 BERT输入嵌入与自注意力.21 2.1.2 通过预测掩码词元训练 BERT.25 2.1.3 将 BERT 微调至下游任务.28 2.1.4 注意力与嵌入的可视化.29 2.1.5 BERT 的自然语言理解.32 2.1.6 计算复杂度.35 2.1.7 总结.36 2.2 GPT：自回归语言模型.37 2.2.1 自回归语言模型的任务.37 2.2.2 通过预测下一个词元训练 GPT.38 2.2.3 生成词语序列.40 2.2.4 先进的语言模型 GPT-2.42 2.2.5 微调 GPT.43 2.2.6 总结.43 2.3 Transformer：序列到序列翻译.44 2.3.1 Transformer 架构.44 2.3.2 通过解码翻译生成词语.48 2.3.3 翻译评估.49 2.3.4 预训练语言模型与基础模型.50

2.3.5 总结.54 2.4 预训练语言模型的训练与评估.55 2.4.1 预训练语言模型优化.56 2.4.2 预训练语言模型正则化.59 2.4.3 神经架构搜索.60 2.4.4 模型预测的不确定性.61 2.4.5 模型预测解释.64 2.4.6 总结.69

3 改进预训练语言模型 .77

3.1 修改预训练目标.78

3.1.1 类似 BERT 的自编码器.79 3.1.2 类似 GPT 的自回归语言模型. . . . . 83 3.1.3 Transformer 编码器-解码器.89 3.1.4 Transformer 变体的系统比较.94 3.1.5 总结.96

3.2 捕获更长依赖关系.97

3.2.1 稀疏注意力矩阵.98 3.2.2 哈希与低秩近似.99 3.2.3 长输入序列下 Transformer 的比较 . . 102 3.2.4 总结.103

3.3 多语言预训练语言模型.104

3.3.1 自编码器模型.104 3.3.2 Seq2seq Transformer模型.107 3.3.3 自回归语言模型.109 3.3.4 总结.109

3.4 预训练语言模型的额外知识.110

3.4.1 利用知识库嵌入.111 3.4.2 用于图学习的预训练语言模型 . . . . . 114 3.4.3 表格的文本编码 .116 3.4.4 知识库关系的文本编码 . 118 3.4.5 利用检索文本增强预训练语言模型 121 3.4.6 总结.122 3.5 改变模型规模.123 3.5.1 更大的模型通常性能更好 . . . . . 124 3.5.2 混合专家模型.125 3.5.3 参数压缩与缩减.128 3.5.4 低秩分解.129 3.5.5 知识蒸馏.129 3.5.6 总结.130 3.6 特定应用的微调.131 3.6.1 微调的性质. 132 3.6.2 微调变体.134 3.6.3 创建少样本提示.137 3.6.4 用于少样本推理学习的思维链 138

3.6.5 微调模型以执行指令.139 3.6.6 利用基础模型生成带标签的数据 143 3.6.7 总结.144

4 基础模型获取的知识 s …… ……157

4.1 基准集合……158

4.1.1 GLUE基准集合……159 4.1.2 SuperGLUE：GLUE 的高级版本……159 4.1.3 文本补全基准……161 4.1.4 大型基准集合…… 162 4.1.5 总结……164

4.2 通过探测分类器评估知识.165 4.2.1 BERT 的句法知识.165 4.2.2 常识知识.166 4.2.3 逻辑一致性.168 4.2.4 总结.171 4.3 基准测试的可迁移性与可复现性.171

4.3.1 基准测试结果的可迁移性.172 4.3.2 自然语言处理领域已发表结果的可复现性.174 4.3.3 总结.175

5 用于信息抽取的基础模型 .181

5.1 文本分类.182 5.1.1 互斥类别的多类分类.184 5.1.2 多标签分类. 186 5.1.3 少样本与零样本分类.188 5.1.4 总结.190 5.2 词义消歧.191 5.2.1 义项集合.191 5.2.2 模型.192 5.2.3 总结.194 5.3 命名实体识别. 195

5.3.1 平面命名实体识别.195 5.3.2 嵌套命名实体识别.196 5.3.3 实体链接.198 5.3.4 总结.200

5.4 关系抽取.201 5.4.1 共指消解.202 5.4.2 句子级关系抽取.203 5.4.3 文档级关系抽取.204 5.4.4 实体与关系联合抽取.204 5.4.5 远程监督. 209 5.4.6 利用布局信息进行关系抽取.210 5.4.7 总结.212

6 文本生成基础模型 .221
6.1 文档检索.222 6.1.1 稠密检索.223 6.1.2 文本检索性能测量.224
6.1.3 使用 BERT 的交叉编码器.225 6.1.4 使用词元嵌入进行检索.227
6.1.5 稠密段落嵌入与最近邻搜索 . . 229 6.1.6 总结.231 6.2 问答.232

6.2.1 基于训练数据知识的问答 . . . 233 6.2.2 基于检索的问答.
236 6.2.3 使用检索的长文本问答.239 6.2.4 总结.242

6.3 神经机器翻译.243
6.3.1 单一语言对翻译.244 6.3.2 多语言翻译.246 6.3.3 多语言问答.
249 6.3.4 总结.251

6.4 文本摘要生成.252
6.4.1 较短文档.253 6.4.2 较长文档.255 6.4.3 多文档摘要.256
6.4.4 总结.257

6.5 文本生成.258
6.5.1 通过语言模型生成文本.260 6.5.2 生成具有给定风格的文本.
262 6.5.3 将文档转换为另一种文本风格.264 6.5.4 根据给定情节
生成故事.267 6.5.5 生成虚假新闻.273 6.5.6 生成计算机代码.276
6.5.7 总结.277

6.6 对话系统.278 6.6.1 作为模块流水线的对话模型.280 6.6.2 高级对话
模型.281 6.6.3 使用检索和过滤器的 LaMDA 与 BlenderBot 3 . . .. . .
284 6.6.4 对话系统的局限性与补救措施.287 6.6.5 总结.289

7 语音、图像、视频与控制基础模型 303
7.1 语音识别与生成.304 7.1.1 自动语音识别基础.305 7.1.2 基于
Transformer的语音识别.306 7.1.3 语音识别自监督学习.307
7.1.4 文本转语音.310 7.1.5 语音到语音语言模型.312

7.1.6 音乐生成.313 7.1.7 总结.314 7.2 图像处理与生成.315 7.2.1 图像处理基础.315 7.2.2 视觉Transformer.317 7.2.3 图像生成.320 7.2.4 文本与图像联合处理.322 7.2.5 文本描述图像.324 7.2.6 从文本生成图像.327 7.2.7 扩散模型从噪声破坏的图像中恢复图像 . . . . . 330 7.2.8 多用途模型.335 7.2.9 总结.337 7.3 视频理解与生成.339 7.3.1 视频处理基础.339 7.3.2 视频描述生成.342 7.3.3 视频动作识别.343 7.3.4 从文本生成视频.350 7.3.5 总结.353 7.4 动力系统控制.354 7.4.1 决策变换器.355 7.4.2 用于文本、图像和控制的GATO模型.357 7.4.3 总结. 359 7.5 DNA与蛋白质序列的解读.360 7.5.1 总结.360

8 总结与展望 .371

8.1 基础模型是一种新范式.372 8.1.1 预训练语言模型.372 8.1.2 基础模型对不同模态进行联合处理.373 8.1.3 基础模型的性能水平.375 8.1.4 前景广阔的经济解决方案.379 8.2 基础模型的潜在危害.381 8.2.1 无意生成有偏见或虚假陈述.383 8.2.2 基础模型造成的蓄意伤害.387 8.2.3 过度依赖或将基础模型视为人工 . . . . . 389 8.2.4 泄露私人信息.389 8.2.5 社会、获取与环境危害.391 8.3 高级人工智能系统.395 8.3.1 基础模型能否生成创新内容？ . . . . . 396 8.3.2 让语言在现实世界中落地. 396 8.3.3 快思考与慢思考.399 8.3.4 规划策略.400

A 附录 .407 A.1 图形中所用图像的来源与版权.407 索引411

索引 · 411

第 1 章 引言

摘要 大约十年前，高效深度学习模型的发展使许多深度神经网络得以用于解决模式识别任务，例如自然语言处理和图像识别。这些模型的一个优势是，它们能够自动生成按层组织的特征，以表示内容，而无需人工构建特征。这些模型依赖于机器学习，运用统计技术赋予机器从数据中“学习”的能力，而无须明确告知它们该做什么。深度学习模型逐层转换输入，使数据中的复杂模式能够被识别。本章首先介绍文本如何经过预处理并划分为词元，这些词元构成自然语言处理的基础。随后，我们概述若干经典的机器学习模型，它们常作为模块用于高级模型中。示例包括逻辑分类器模型、全连接层、循环神经网络和卷积神经网络。

关键词： 自然语言处理、文本预处理、向量空间模型、静态嵌入、循环网络、卷积网络

1.1 本书范围

大约十年前，高效深度学习模型的发展，使许多深度神经网络得以用于解决模式识别任务，例如自然语言处理 （自然语言处理）和图像处理。通常，模型必须捕捉文本或图像的意义，并做出适当的决策。或者，它们也可以根据手头的任务生成新的文本或图像。这些模型的优势在于，它们会生成按层排列的中间特征，无需人工构造特征。诸如卷积神经网络（卷积神经网络） [32] 和循环神经网络

（循环神经网络） [65]使用低维稠密向量作为一种分布式表示，以表达语言的句法和语义特征。

所有这些模型都可以视为人工智能 （*AI*）系统。人工智能是一个广泛的研究领域，旨在创造能够像具有自然智能的人类和动物一样行动的智能机器。它体现了该领域的长期目标，即构建能够模仿并最终超越人类全部认知能力的机器。机器学习（*ML*）是人工智能的一个子领域，它采用统计技术，使机器能够从数据中“学习”，而无需获得关于应执行何种操作的明确指令。这一过程也称为“训练”，在此过程中，“学习算法”会逐步提升模型在给定任务上的性能。深度学习是机器学习的一个领域，其中输入会逐层转换，以便识别数据中的复杂模式。“深度”这一形容词指的是现代机器学习模型中大量的层，这些层有助于学习数据的表达性表示，从而实现更好的性能。

与计算机视觉相比，NLP 应用所需的文本标注训练数据规模相当小，仅包含几千个句子（机器翻译除外）。其主要原因是人工标注成本高昂。为了避免过拟合，即模型过度适应随机波动，只能训练相对较小的模型，因此无法获得很高的性能。在过去五年中，研究人员基于 Vaswani 等人提出的 *Transformer* [67]开发了新的自然语言处理方法。它们通过一个由实数构成的向量（称为嵌入）来表示每个词语。在这些嵌入之间可以计算各种“注意力”，这些注意力可以被视为不同词语之间的一种“相关性”。在网络的更高层中，注意力计算被用于生成新的嵌入，从而捕捉词语意义中的细微差别。尤其是，这些嵌入能够根据上下文理解同一个词语所产生的不同意义。这些模型的一项关键优势是可以使用未标注文本进行训练，而这类文本几乎可以无限获取，因此不会出现过拟合问题。

目前，该研究领域的新方法发展迅速，因此许多早些年的方法已经过时。这些模型通常分两个步骤进行训练：首先，在预训练 阶段，模型在包含数十亿词语且未经任何标注的大型文本语料库上进行训练。一种典型的预训练任务是预测输入中被掩码的单个词语。通过这种方式，模型能够学习自然语言句法和语义的细微之处。由于可用数据充足，模型可以扩展到包含数百万或数十亿个参数的许多层。

在第二个微调步骤中，模型会使用一个规模较小的带注释训练集进行训练。这样，模型便能适应新的特定任务。由于微调数据与预训练数据相比非常少，而且模型容量很大，拥有数百万个参数，因此它能够适应微调任务，同时保留已存储的语言结构信息。研究表明，这一理念可应用于大多数自然语言处理任务，从而在语义理解方面带来前所未有的性能提升。这种迁移学习使知识能够从预

训练阶段迁移到微调模型中。这些模型称为预训练语言模型（预训练语言模型）。

近年来，这些预训练语言模型的参数数量随着更多训练数据的加入而系统性地增加。事实证明，与传统认知相反，这些模型的性能越来越好，并未受到过拟合的影响。如果给予一些起始文本作为提示，拥有数十亿参数的模型能够生成句法正确、语义一致且流畅的文本。它们可以回答问题，并对不同类型的提示作出有意义的反应。

此外，同一种预训练语言模型架构可以同时针对不同类型的序列进行预训练，例如文本中的词元、图片中的图像块、语音的声音片段、视频帧中的图像块序列、DNA片段等。它们能够同时处理这些媒体类型，并在不同模态之间建立联系。通过自然语言提示，可以调整它们，使其在各种任务上达到可接受的表现，即使它们并未针对这些任务接受过明确训练。由于这种灵活性，这些模型很有希望被用于开发跨领域应用。因此，拥有数十亿参数的大型预训练语言模型通常被称为基础模型 [9]。

本书旨在全面介绍当前预训练语言模型和基础模型的最新进展，重点关注其在自然语言处理中的应用：

- 我们介绍必要的背景知识、模型架构、预训练和微调任务，以及评估指标。
- 我们讨论各类自然语言处理应用中最相关的模型，这些模型目前具有最佳的准确率或性能，也就是说，接近最先进水平（最先进水平）。我们的目的不是罗列近年来开发的所有模型，而是讲解一些有代表性的模型，使读者能够理解其内部工作原理。
- 近年来，预训练语言模型已应用于多种语音、图像和视频处理任务，由此产生了“基础模型”这一术语。我们概述了最相关的模型；这些模型通常支持不同媒体（例如文本和图像）的联合处理
- 我们提供可用模型代码和预训练模型参数的链接。
- 我们讨论这些模型的优势和局限，并展望未来可能的发展。

此前已有不少关于深度学习和自然语言处理的综述 [1–4, 10, 15, 16, 27, 39, 50, 53, 54, 59, 66]。Hanet al. [22]，、Lin等人 [41]，和 Kalyan et al. [31] 的综述是最新、最全面的。Jurafsky 和 Martin [30]正在编写一本关于这一领域的最新著作。此外，还有许多针对特定模型变体或应用领域的综述。在适当之处，我们会提供这些综述的参考文献。新术语通常以斜体 印出，模型则以**粗体**印出。

本章其余部分介绍文本预处理和经典自然语言处理模型，其中一部分会在预训练语言模型中得到复用。第二章介绍预训练语言模型的主要架构，这些模型目前是自然语言处理领域的主力。第三章介绍大量扩展基础模型能力的预训练语言模型变体。第四章介绍信息

预训练语言模型和基础模型所捕获的信息，并分析它们的句法能力、世界知识和推理能力。

本书接下来将考察各种应用领域，并确定目前在每个领域中以合理成本提供最佳结果的预训练语言模型和基础模型。第五章介绍能够自动识别文本文件中的结构化信息和语言特征的信息抽取方法，例如用于关系抽取的方法。第六章讨论能够自动生成自然语言新文本的自然语言生成方法，通常用于响应提示。第七章专门介绍用于分析和创建多模态内容的模型，这些模型通常会整合两种或更多模态之间的内容理解与生产，例如文本、语音、图像、视频等。总体趋势是，更多的数据、更强的计算能力以及更大的参数集会带来更好的性能。最后的总结章节对此进行了说明，同时还考察了基础模型的社交和伦理方面，并总结了未来可能的发展方向。

1.2 文本预处理

预处理的第一步是提取实际文本。对于每种类型的文本文件，例如 PDF、HTML、XML、DOCX 和 ePUB，都有专用的解析器，可将文本解析为字符、词语和格式信息。通常会移除布局和格式信息。

然后，提取出的文本通常会被划分为词元，即单词、数字和标点符号。这个过程并不简单，因为文本通常包含电话号码或电子邮件地址等特殊单元，必须采用特殊方式进行处理。一些文本挖掘任务要求将文本切分为句子。不同语言的分词器和句子切分器已在过去几十年中得到开发，并且可以从许多编程工具箱中调用，例如*spaCy* [64]。

过去，许多预处理方法旨在生成新的相关特征（词性标注、句法分析树）并移除不必要的词元（词干提取、停用词去除、词形还原）。在大多数情况下，采用现代方法后，这已不再必要，因为现代方法会在内部自动提取与当前任务相关的特征。

在可选的最后一步中，单词词元还可以进一步细分和重新排列。一种简单的技术是创建字符 n-*gram*（即单词中所有由 n 个相邻字符组成的序列），作为附加特征。或者，也可以构成词语 n-*gram*，其中包含 n 个连续的词语。

目前，最流行的方法是尝试限制词汇表中不同单词的数量。常见的选择是字节对编码 [19]。该方法首先将所有角色选为词元。然后，依次将出现频率最高的词元对合并为一个新词元，并将所有词元对实例替换为新词元。重复这一过程，直到获得指定大小的词汇表。请注意，新单词始终可以表示为一系列词汇表词元，并且

表 1.1 自然语言处理模型中使用的文档表示。

Type	**由.....生成**	**用于.....**
词袋 简单嵌入 上下文嵌入	分词与计数 相关与回归：主题模型 [7], Word2Vec [46], GloVe [51]。 注意力计算：ElMo [52], Transformer [67], GPT [55], BERT [17] 以及许多其他模型。	逻辑分类器、支持向量机。第 1.3 节。 分类器、聚类、可视化，循环神经网络等。第1.5节 利用监督训练进行微调数据。第2.1节。

角色。常见单词最终会成为词汇表的一部分，而较罕见的单词则会被拆分成组件，这些组件通常保留某些语言学意义。这样便可以避免出现词汇表外词语。

WordPiece [69] 算法也从将该集合中的所有字符选作词元开始。然后，它假设文本语料库是按照词元的观测频率随机采样生成的。它以使训练数据的似然最大幅度提高 [60]的方式合并词元 a 和 b（在单词内部）。存在一种快速变体，其计算复杂度相对于输入长度呈线性关系 [63]。*SentencePiece* [35]是一个包含多种子词分词器的软件包，也可应用于所有亚洲语言。所有这些方法实际上都在高频词的词级输入与低频词的字符级输入之间进行插值。

通常必须确定输入文本的语言 [29, 57]。大多数语言识别方法会从输入文本中提取字符 n-gram，并评估其相对频率。一些方法可应用于同时包含不同语言的文本 [42, 71]。要从文本中滤除冒犯性词语，可以使用不同语言中的此类有害词语列表 [62]。

1.3 向量空间模型与文档分类

要将机器学习应用于文档，必须将其中的文本转换为标量、向量、矩阵或更高维的数字排列，这些统称为 张量。在上一节中，语料库中的文本文件通过预处理被转换为词元序列。现在，这些词元必须被转换为张量。

词袋表示通过词元计数的向量 d 来描述给定的文本文件$\boldsymbol{x}$ 。词汇表是训练文档集合（即训练语料库）中包含的所有不同词元的列表。不考虑词元的顺序，这个词袋向量记录了词汇表中的每个词元在文档 d中出现的频率。请注意，大多数向量条目都将为零，因为每个文档只会包含词汇表词元中的一小部分。计数向量可以经过修改

以突出显示信息量较高的词元，例如使用*tf-idf* 统计量[43]。表 1.1总结了用于自然语言处理的不同文档表示。

文档分类方法旨在根据文本文件的内容对其进行分类 [33, 61]。一个重要示例是逻辑分类器，它将词袋向量 $\boldsymbol{x}$ 作为输入，并预测每个 k 个可能输出类别 $y \in \{1, \ldots, k\}$的概率。更准确地说，存在一个随机变量 Y，其取值可能为 $1, \ldots, k$。为了根据输入 $\boldsymbol{x}$预测输出类别 y，首先生成一个得分向量，形式如下

$$\boldsymbol{u} = A\boldsymbol{x} + \boldsymbol{b} \tag{1.1}$$

通过对输入 $\boldsymbol{x}$进行仿射变换。这里，向量 $\boldsymbol{x}$ 经过线性变换 $A\boldsymbol{x}$，然后加上一个偏见向量 $\boldsymbol{b}$。随后，将长度为 k 的结果得分向量 $\boldsymbol{u}$ 通过*softmax* 函数转换为在 k个类别上的概率分布

$$\text{softmax}(u_1, \ldots, u_k) = \frac{(\exp(u_1), \ldots, \exp(u_k))}{\exp(u_1) + \cdots + \exp(u_k)}, \tag{1.2}$$

$$p(Y{=}m|\boldsymbol{x}; A, \boldsymbol{b}) = \text{softmax}(A\boldsymbol{x} + \boldsymbol{b}). \tag{1.3}$$

由于 softmax 函数会将任意向量转换为概率向量，因此我们得到输出类别 m 关于输入 $\boldsymbol{x}$的条件概率函数。该函数

$$\text{LRM}(\boldsymbol{x}) = \text{softmax}(A\boldsymbol{x} + \boldsymbol{b}) \tag{1.4}$$

称为带参数向量 $\boldsymbol{w} = vec(A, b)$ 的逻辑分类器模型 [48]。一般而言，将输入 $\boldsymbol{x}$ 映射到输出 y 或输出上的概率分布的函数，称为模型 $f(\boldsymbol{x}; \boldsymbol{w})$。

模型使用训练数据 $Tr = \{(\boldsymbol{x}^{[1]}, y^{[1]}), \ldots\ldots, (\boldsymbol{x}^{[N]}, y^{[N]})\}$进行训练，其中示例 $(\boldsymbol{x}^{[i]}, y^{[i]})$必须独立同分布（独立同分布）。任务是调整参数 $\boldsymbol{w}$，使预测概率$p(Y{=}m|\boldsymbol{x}; \boldsymbol{w})$最大化。根据最大似然原理，可以通过修改参数向量 $\boldsymbol{w}$，使完整训练数据具有最大的概率 [24,p来实现。 31]

$$\max_{\boldsymbol{w}} = p(y^{[1]}|\boldsymbol{x}^{[1]}; \boldsymbol{w}) * \cdots * p(y^{[N]}|\boldsymbol{x}^{[N]}; \boldsymbol{w}). \tag{1.5}$$

对表达式取对数并乘以 –1.0，可得到分类损失函数 $L_{\text{MC}}(\boldsymbol{w})$，也称为最大熵损失。

$$L_{\text{MC}}(\boldsymbol{w}) = -\left[\log p(y^{[1]}|\boldsymbol{x}^{[1]}; \boldsymbol{w}) + \cdots + \log p(y^{[N]}|\boldsymbol{x}^{[N]}; \boldsymbol{w})\right]. \tag{1.6}$$

为了优化损失函数，需要计算其梯度，并通过随机梯度优化或其他优化器将其最小化（参见第 2.4.1 节）。

分类器的性能通过准确率、精确率、召回率、F1值等指标，在单独的测试数据上进行测量。 [21, 第410页及后页]。由于词袋表示忽略了重要的词序信息，使用逻辑

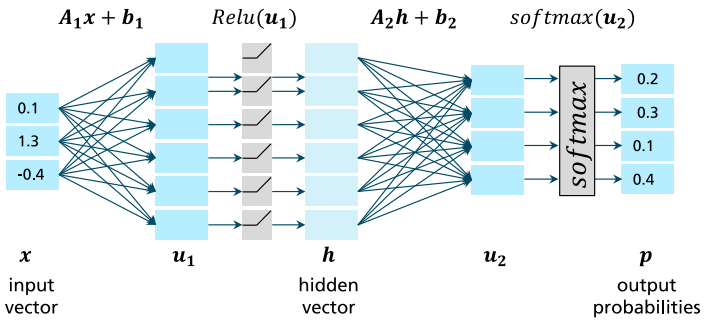


图 1.1 用于分类的神经网络通过包含仿射变换和非线性激活函数（例如 ReLU）的层对输入进行变换。最后一层通常是逻辑分类器。

分类器进行文档分类如今已不太常见。然而，该模型仍然是大多数深度学习架构的组成部分。

1.4 非线性分类器

事实证明，逻辑分类器通过线性超平面对输入空间进行划分，而这些线性超平面无法解决更复杂的分类任务，例如异或问题[47]。一种替代方案是通过额外的仿射变换 $\boldsymbol{h}$，随后应用单调非递减的非线性激活函数 g，生成一个内部隐藏向量 $\boldsymbol{h}$，并将该隐藏向量作为逻辑分类器的输入，以预测随机变量 Y

$$\boldsymbol{h} = g(A_1\boldsymbol{x} + \boldsymbol{b}_1), \tag{1.7}$$

$$p(Y{=}m|\boldsymbol{x};\boldsymbol{w}) = \text{softmax}(A_2\boldsymbol{h} + \boldsymbol{b}_2), \tag{1.8}$$

其中，该模型的参数可以收集到参数向量 $\boldsymbol{w} = vec(A_1, b_1, A_2, b_2)$ 中。非线性激活函数 g 的形式相当任意，通常使用 $\tanh(x)$ 或修正线性单元 $\text{ReLU}(x) = \max(0, x)$。$\text{Fcl}(\boldsymbol{x}) = g(A_1\boldsymbol{x} + \boldsymbol{b}_1)$ 称为全连接层 。

该模型（图 1.1）能够以任意高的精度解决任何分类问题，前提是 $\boldsymbol{h}$ 的长度足够大（[21, 第 192] 页）。通过在网络前面添加更多全连接层，我们得到一个深度神经网络，与浅层网络相比，它只需更少的参数即可逼近更复杂的函数。从历史上看，它被称为多层感知器（MLP）。Liang 等人 [40] 证明，对于一大类分段光滑函数，在给定函数逼近程度的情况下，浅层网络为逼近某个函数所需的隐藏向量大小，比深度网络所需的相应神经元数量大得多，且呈指数级增长。

支持向量机 [14] 采用不同的方法，试图创建一个超平面，使其位于输入空间中两个类别的训练样本之间。此外，该超平面与样本之间还应保持较大的距离（间隔）。即使输入变量的数量较多，例如在文档分类中[28]，该模型也能减少过拟合，并且通常具有较高的分类准确率。后来，人们将其扩展到不同的核损失准则，例如图核 [56] ，其中包括语法特征。除了支持向量机之外，还使用了许多其他分类器，例如随机森林 [24, 第 588 页]和梯度提升树 [24,第 360]，页，它们是最常用的分类器之一。

对于这些传统分类器，分析人员通常必须手动构造输入特征。用于文本分析的现代分类器能够自动创建相关特征（第 2.1 节）。对于自然语言处理模型的训练，主要存在三种范式：

- 监督训练 基于由输入 $\boldsymbol{x}$（例如文档文本）和输出 $\boldsymbol{y}$组成的训练数据对（$\boldsymbol{x}$， $\boldsymbol{y}$），其中 $\boldsymbol{y}$ 通常是人工标注，例如情感。通过优化，模型的未知参数会被调整，以便以最优方式根据输入预测输出。
- 无监督训练 仅考虑一些数据 $\boldsymbol{x}$ ，并从未标注数据中推导出一些内在知识，例如簇、密度或潜在表示。
- 自监督训练 将观测数据向量的一部分选作输入 $\boldsymbol{x}$和输出 $\boldsymbol{y}$。其核心思想是以监督方式根据 $\boldsymbol{x}$ 预测 $\boldsymbol{y}$ 。例如，语言模型是一项自监督任务，尝试根据之前的词元 v_1，……， v_t预测下一个词元 v_{t+1} 。对于自然语言处理模型，这种训练类型的使用非常广泛。

1.5 生成静态词嵌入

词袋词表示的一个问题是，词元的频率向量无法捕捉词语之间的关系，例如同义关系和同音异义关系，也无法反映它们的语义相似度。一种替代方案是采用更具表达力的词语和文档表示，其基础是由 Zellig Harris [23] 和 John Firth [18]推广的分布语义学[58], 理念。按照 Firth 的说法，“一个词的特征取决于它周围的词”。这表明，出现在同一邻域中的词语往往具有相似的意义。

基于这一想法，每个单词都可以用一个 d_{emb} 维向量嵌入(单词) $\in \mathbb{R}^{d_{emb}}$来表示，即一种词嵌入。通常， d_{emb}取值为 100 到 1,000。这些嵌入的构建方式应确保在相似上下文中出现的词语具有距离较小的嵌入向量，例如欧几里得距离。这样，文档就可以表示为由此类嵌入构成的序列。事实证明，如果词语的嵌入距离较小，它们通常具有相似的意义。嵌入可以用作下游文本挖掘任务的输入，例如情感分析。Goldberg [20] 对静态词嵌入作了出色的介绍

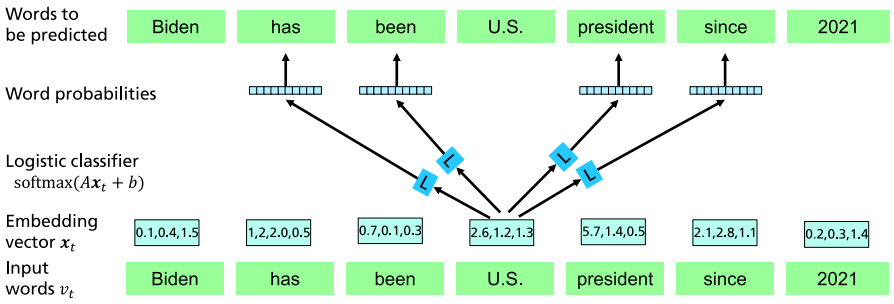


图 1.2 Word2vec 通过逻辑分类器 L预测中心词邻域中的词语。 L 的输入是中心词的嵌入。通过使用大量文档进行训练，可以学习 L 的参数以及嵌入 [54, 第 2] 页。

。这类嵌入称为静态嵌入，因为每个单词只有一个嵌入，且不受上下文影响。

有多种不同的方法可以通过无监督方式生成词嵌入。Collobert 等人 [13] 表明，通过预测邻近词语获得的词嵌入可以提升下游任务的性能，例如命名实体识别和语义角色标注。

Word2vec [45]使用一个极其简单的模型来预测中心词邻域中的词语。如图 1.2 所示，它将中心词的嵌入向量作为逻辑分类器 (1.3) 的输入，以推断其邻域中约五到七个位置处词语的概率。训练目标是以较高的概率预测训练集中的所有邻近词语。在训练过程中，Word2Vec 对语料库中的所有词语重复进行这种预测，并通过随机梯度下降优化逻辑分类器的参数以及嵌入的值，从而提升对邻近词语的预测效果。

一个文本集合的词汇表包含 k 个不同的词语，例如 $k = 100,000$。要通过 Softmax (1.2) 预测第 i个词的概率，必须计算 k 个指数项 $\exp(u_i)$。为了避免这项计算，近似将该分数表示为

$$\frac{\exp(u_i)}{\exp(u_1) + \cdots + exp(u_k)} \approx \frac{\exp(u_i)}{\exp(u_i) + \sum_{j \in S} exp(u_j)}, \tag{1.9}$$

其中， S 是一个小样本，例如从词语索引中随机选取的 10 个索引。这种技术称为噪声对比估计 [21,p. 612]。这种方法有多种变体，几乎适用于所有涉及多类别 Softmax 计算的分类任务。由于随机梯度下降本身使用带噪声的梯度，近似 Softmax 函数所引入的额外噪声并无害处，甚至还能帮助模型摆脱局部极小值。Word2Vec 的浅层架构在表示学习方面，效率远高于以往的架构。

Word2Vec 的嵌入已用于许多下游任务，例如文档分类。此外，只需搜索嵌入与目标词的嵌入之间欧几里得距离较小的词语，就能找出意义相近的词语。例如，与“中子”最接近的邻居是“中子”、“质子”、“氘”、“正电子”和“衰变”。这样便能找出同义词。将嵌入投影到二维空间，可用于对语料库内容进行探索性分析。**GloVe** 利用语料库中的全局词-词共现统计数据，生成类似的嵌入向量 [51]。

事实证明，嵌入之间的差异往往可以作出解释。例如，vec(德国) $-vec$(柏林) $+vec$(巴黎)的结果是，vec(法国)是其欧几里得距离最近邻。这种性质称为类比，在许多关系（如首都-国家、货币-国家等）的大多数示例中都成立 [45]。

FastText [8]表示通过利用子词信息来丰富静态词嵌入。从每个单词中提取指定长度范围（例如 3-6）的字符 n-gram。然后，为单词及其字符 n-gram定义嵌入向量。训练嵌入时，对中心单词邻域内所有单词和字符 n-gram的嵌入求平均，并由逻辑分类器预测中心单词及其字符 n-gram出现的概率。为提升概率预测效果，模型参数通过随机梯度下降进行优化。对训练语料库中的所有单词重复这一过程。训练完成后，可以仅利用未见过单词的 n-gram嵌入来重构这些单词。*Starspace* [68] 是作为 FastText 的泛化形式提出的。它通过分析与任意实体（如作者、产品）相关的文本并评估图结构，从而为这些实体生成嵌入。另一种方法是球面嵌入，它将无监督的单词嵌入和段落嵌入约束在超球面上[44]。

1.6 循环神经网络

循环神经网络 用于对长度不一的序列 $v_1, \ldots, v_T$ 进行建模 T，例如文本文件中的词元序列。考虑这样一个任务：给定之前的词元（$v_1, \ldots, v_t$），预测下一个词元 v_{t+1}。Bengio 等人 [6] 提出，每个词元 v_t 都用一个嵌入向量 $\boldsymbol{x}_t = emb(v_t)$表示，以指示其意义 v_t。之前的词元由一个隐藏向量 $\boldsymbol{h}_t$表征，该向量描述子序列（$v_1, \ldots, v_{t-1}$）的状态。RNN 是一个函数 $\text{Rnn}(\boldsymbol{h}_t, \boldsymbol{x}_t)$，通过以下方式预测下一个隐藏向量 $\boldsymbol{h}_{t+1}$

$$\boldsymbol{h}_{t+1} = \text{RNN}(\boldsymbol{h}_t, \boldsymbol{x}_t). \tag{1.10}$$

随后，逻辑分类器 (1.3)使用参数 H和 $\boldsymbol{g}$，根据h_{t+1}中包含的信息，预测下一个词元 v_{t+1} 的概率向量，

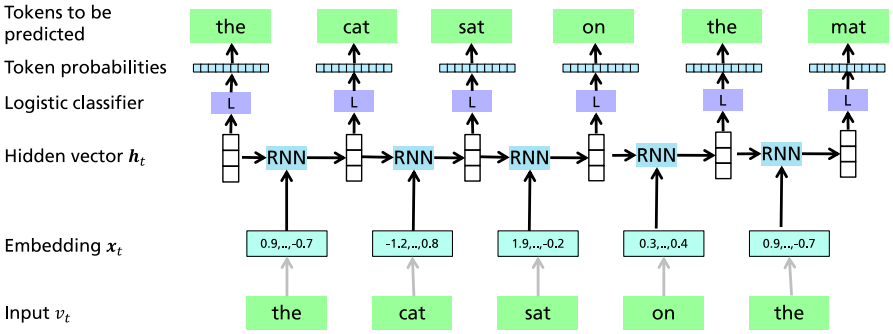


图 1.3 循环神经网络从左侧开始，以之前的词元为条件，使用逻辑分类器依次预测下一个词元的概率 L。隐藏向量 $\boldsymbol{h}_t$ 存储位置 t之前出现的词元信息。

$$p(V_{t+1}|v_1,\ldots,v_t) = \text{softmax}(H * \boldsymbol{h}_{t+1} + \boldsymbol{g}), \tag{1.11}$$

如图 1.3 所示。这里， V_t 是位置 t处可能出现的词元这一随机变量。根据条件概率的定义，整个序列的联合概率可以分解为

$$p(v_1,\ldots,v_T) = p(V_T = v_T|v_1,\ldots,v_{T-1}) * \cdots * p(V_2 = v_2|v_1) * p(V_1 = v_1). \tag{1.12}$$

能够计算自然语言文本联合概率或条件概率的模型称为语言模型，因为它可能涵盖有关该语言的所有信息。根据条件概率依次预测下一个单词的语言模型通常称为自回归语言模型。根据（1.12），可以将观察到的词元（$v_1,\ldots,v_t$）作为输入，来预测下一个词元的概率 V_{t+1}。这些概率的乘积即为观察到的词元序列（$v_1,\ldots,v_T$）的正确联合概率。相同的模型 $\text{Rnn}(\boldsymbol{h},\boldsymbol{x})$会被反复应用，并生成一系列隐藏向量$\boldsymbol{h}_t$。简单循环神经网络仅由一个全连接层组成

$$\text{Rnn}(\boldsymbol{h}_t,\boldsymbol{x}_t) = \tanh\left(A * \begin{bmatrix}\boldsymbol{h}_t \\ \boldsymbol{x}_t\end{bmatrix} + \boldsymbol{b}\right). \tag{1.13}$$

预测词语的概率 $v_1,\ldots,v_T$取决于参数$\boldsymbol{w} = vec(H,\boldsymbol{g},A,\boldsymbol{b}$, 词元嵌入$(v_1),\ldots,$词元嵌入$(v_T))$。为了提高这些概率，我们可以使用随机梯度下降优化器（第 2.4.1 节），并调整 $\boldsymbol{w}$ 中的未知参数。请注意，这还包括对新词元嵌入词元嵌入(v_t)的估计。 [70, 第 8-9] 章给出了近期综述。

事实证明，该模型难以重建相距较远的序列元素之间的关系，因为随着序列变长，梯度往往会消失或“爆炸”。因此，人们开发出了新的循环神经网络类型，例如长

短期记忆网络 （LSTM） [26] 和门控循环单元 （GRU） [11], ，它们能够更好地捕捉序列中的长程依赖关系。

除了预测序列中的下一个单词，循环神经网络还成功应用于预测序列元素的性质，例如命名实体识别 [36]和关系抽取 [38]。针对这些应用，研究人员开发了双向循环神经网络，它由前向语言模型和后向语言模型组成。前向语言模型 从文本开头开始，预测下一个词元；而后向语言模型 从文本末尾开始，预测前一个词元。双向长短期记忆网络也称为双向长短期记忆网络。此外，研究人员还提出了多层循环神经网络 [72], ：某一层的循环神经网络单元生成的隐藏向量会作为下一层循环神经网络单元的输入，而最后一层则提供当前任务的预测结果。

机器翻译 是循环神经网络的一项重要应用，可将一种语言翻译成另一种语言 [5]。在这个过程中，首先由编码器 循环神经网络将输入句子编码为一个隐藏向量 $\boldsymbol{h}_T$。随后，第二个解码器 循环神经网络将这个隐藏向量用作初始隐藏向量，以生成目标语言句子的词语。然而，循环神经网络仍难以捕捉序列元素之间的长距离关系，因为它无法覆盖相距较远的序列元素之间的直接关系。

注意力 最初用于机器翻译领域，以便跨越长距离传递信息。它计算解码器循环神经网络的隐藏向量与编码器循环神经网络在不同位置的隐藏向量之间的相关性。利用这种相关性，可以将相关的编码器隐藏向量加权平均，构建一个上下文向量 。然后，利用该上下文向量来改进最终的翻译结果 [5]。由此产生的翻译远优于原始循环神经网络生成的翻译。我们将在后续章节看到，注意力是构建更优自然语言处理模型的一项基本原则。

ELMo [52] 使用多层双向LSTM语言模型生成嵌入。该模型使用大型未标注文本语料库，作为前向和后向语言模型进行预训练。在微调过程中，利用隐藏向量的平均值，根据带注释的训练集预测单词的性质。这些语言模型会考虑某个位置前后的词语，因此会对位于中心位置的单词采用上下文表示。对于情感分析、问答和文本蕴含等多种任务，ELMo 能够提升最先进水平的性能。

1.7 卷积神经网络

卷积神经网络（卷积神经网络） [37] 因其在图像领域的出色表现而广为人知。它们以一小块正方形参数排列开始，这些参数称为滤波器内核，并在图像的输入像素矩阵上移动。滤波器内核的值与底层像素值相乘，生成一个输出值。对输入像素矩阵的每个位置都重复这一过程。在

训练期间，滤波器内核的参数会自动调整，以便检测斑块或线条等局部图像模式。网络的每一层（也称为卷积层）都包含许多滤波器内核，而一个网络包含若干卷积层。其间穿插的最大池化层通过取最大值对像素进行局部聚合。卷积神经网络的最后一层通常是带有 Softmax分类器的全连接层。

他们的突破性成果是 *AlexNet* [34]，该模型接收图像的 RGB像素矩阵作为输入，其任务是为图像分配一个内容类别。该模型赢得了 2012年 *ImageNet* 竞赛；在该竞赛中，图像必须被分配到 1000 个类别中的一个，并且该模型展现了深度神经网络的卓越性能。更早之前，Cireşan等人 [12] 的深度卷积神经网络在多个图像分类基准上取得了最先进水平的性能。一个非常成功的卷积神经网络是 *ResNet* [25]，它采用了一种充当旁路的所谓 残差连接 。该连接可以绕过训练初期的许多层，是训练具有数百层的神经网络的关键。由此得到的图像分类器，其准确率高于人类。

虽然循环神经网络被认为是处理文本等序列输入的最佳方式，但人们也提出了一些基于CNN的架构，并在一些自然语言处理任务上取得了很高的性能。Kim [32] 提出了一种相对浅层的用于句子分类的卷积神经网络。它包含一个嵌入层、一个卷积层、一个最大池化层，以及一个带有 Softmax输出的全连接层。一维卷积 被应用于输入词语的嵌入，基本上是将相邻词语中存储的信息组合起来，并将它们视为 n-gram。嵌入会通过带有可训练权重的移动平均进行处理。事实证明，使用这种架构进行分类非常高效，其性能与更难训练的循环架构相近。

另一种有趣的卷积神经网络架构是 *WaveNet* [49]，，这是一种更深层的网络，主要用于文本到语音合成。它由多个层层堆叠的卷积层组成，其核心是扩张因果卷积。因果是指，在位置 t 处的卷积只能利用先验信息 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}$。扩张是指，卷积可以按一定步长跳过输入值 k，也就是说，在某一层中，位置 t 处的特征是利用位置 $t, t-k, t-2k, \ldots$ 处的信息进行预测的。这个步长 k 在后续每一层中都会翻倍，因此扩张率的大小依次为 $k^0, k^1, k^2, \ldots$。通过这种方式，可以将非常长的时间跨度纳入预测。这种模型架构已被证明在文本到语音合成方面能够取得很好的结果。

1.8 总结

传统自然语言处理拥有悠久的历史，机器学习模型已在该领域使用了数十年。它们都需要进行一些预处理步骤，以便从输入文本中生成单词或词元。词元尤其有价值，因为它们构成了一个有限大小的字典，并允许通过组合来表示任意单词。

因此，大多数预训练语言模型都会使用它们。早期的文档表示方式（如词袋）如今已经过时，因为它们忽略了序列信息。然而，基于这些表示方式的分类器（如逻辑分类器和全连接层）是预训练语言模型的重要构建模块。

静态词嵌入的概念开启了自然语言处理领域的革命，而自然语言处理建立在上下文词嵌入的基础上。这些思想将在下一章中详细阐述。循环神经网络曾被用于实现首个成功的语言模型，但后来完全被基于注意力的模型取代。用于图像处理的卷积神经网络仍广泛应用于许多应用中。如今，预训练语言模型在图像数据上的性能通常也相近；正如第7章所述，有时会将卷积神经网络与预训练语言模型结合起来，以发挥各自的优势。

参考文献

[1] M. Allahyari、S. Pouriyeh、M. Assefi、S. Safaei、E. D. Trippe、J. B. Gutierrez 和 K. Kochut。《文本挖掘简要综述：分类、聚类与抽取技术》。2017年。arXiv：1707.02919。[2] M. Z. Alom 等人。《深度学习理论与架构的最新进展综述》。发表于：*Electronics* 8.3（2019年），第292页。[3] M. Z. Alom 等人。《历史始于AlexNet：深度学习方法综合综述》。2018年。arXiv：1803.01164。[4] Z. Alyafeai、M. S. AlShaibani 和 I. Ahmad。《自然语言处理中的迁移学习综述》。2020年。arXiv：2007.04239。[5] D. Bahdanau、K. Cho 和 Y. Bengio。《通过联合学习对齐与翻译实现神经机器翻译》。2014年。arXiv：1409.0473。[6] Y. Bengio、R. Ducharme、P. Vincent 和 C. Jauvin。《神经概率语言模型》。发表于：*J. Mach. Learn. Res.* 3（2003年2月），第1137–1155页。[7] D. M. Blei。《概率主题模型导论》。发表于：《*ACM*通讯》55.4（2011年），第77–84页。[8] P. Bojanowski、E. Grave、A. Joulin 和 T. Mikolov。《利用子词信息丰富词向量》。发表于：计算语言学协会会刊 5（2017年），第135–146页。[9] R. Bommasani 等人。《基础模型的机遇与风险》。2021年。arXiv：2108.07258。[10] J. Chai 和 A. Li。《自然语言处理中的深度学习：最新进展综述》。发表于：*2019 Int. Conf. Mach. Learn. Cybern. ICMLC*。《2019年国际机器学习与控制论会议（ICMLC）》。2019年7月，第1–6页。doi：10.1109/ICMLC48188.2019.8949185。[11] J. Chung、C. Gulcehre、K. Cho 和 Y. Bengio。《门控循环神经网络在序列建模中的实证评估》。2014年。arXiv：1412.3555。[12] D. Cireşan、U. Meier 和 J. Schmidhuber。《用于图像分类的多列深度神经网络》。2012年2月13日。arXiv：1202.2745。[13] R. Collobert、J. Weston、L. Bottou、M. Karlen、K. Kavukcuoglu 和 P. Kuksa。《几乎从零开始的自然语言处理》。发表于：*J. Mach. Learn. Res.* 12（2011年），第2493–2537页。[14] C. Cortes 和 V. Vapnik。《支持向量网络》。发表于：*Mach. Learn.* 20.3（1995年），第273–297页。[15] M. Danilevsky、K. Qian、R. Aharonov、Y. Katsis、B. Kawas 和 P. Sen。《自然语言处理中可解释人工智能现状综述》。2020年。arXiv：2010.00711。

[16] S. Dargan、M. Kumar、M. R. Ayyagari 和 G. Kumar。《深度学习及其应用综述：机器学习的新范式》。发表于：Arch. Comput. Methods Eng.（2019年），第1–22页。[17] J. Devlin、M.-W. Chang、K. Lee 和 K. Toutanova。《BERT：面向语言理解的双向 Transformer 预训练》。2018年。arXiv：1810.04805。[18] J. R. Firth。《语言学理论概要，1930–1955，1952–59卷》。发表于：Philol. Soc.（1957年）。[19] P. Gage。《一种数据压缩新算法》。发表于：C Users J. 12（1994年2月1日）。[20] Y. Goldberg。《自然语言处理神经网络模型入门》。发表于：《人工智能研究杂志》57（2016年），第345–420页。[21] I. Goodfellow、Y. Bengio 和 A. Courville。《深度学习》。第1卷。麻省理工学院出版社，剑桥，2016年。网址：https://www.deeplearningbook.org/。[22] X. Han 等人。《预训练模型：过去、现在与未来》。发表于：AI Open（2021年8月26日）。ISSN：2666-6510。doi：10.1016/j.aiopen.2021.08.002。[23] Z. S. Harris。《分布结构》。发表于：Word 10.2-3（1954年），第146–162页。[24] T. Hastie、R. Tibshirani 和 J. Friedman。《统计学习要素：数据挖掘、推理与预测》。第2版，修正后的第12次印刷。施普林格科学与商业媒体，2017年。网址：https://web.stanford.edu/~hastie/Papers/ESLII.pdf。[25] K. He、X. Zhang、S. Ren 和 J. Sun。《用于图像识别的深度残差学习》。发表于：IEEE计算机视觉与模式识别会议论文集。2016年，第770–778页。[26] S. Hochreiter 和 J. Schmidhuber。《长短期记忆》。发表于：Neural Comput. 9.8（1997年），第1735–1780页。[27] A. Hotho、A. Nürnberger 和 G. Paaß。《文本挖掘简要综述》。发表于：Ldv Forum。第20卷，第1期。2005年，第19–62页。[28] T. Joachims。《使用支持向量机进行文本分类：利用大量相关特征进行学习》。发表于：欧洲机器学习会议。施普林格，1998年，第137–142页。[29] A. Joulin、E. Grave、P. Bojanowski 和 T. Mikolov。《高效文本分类的技巧集》。2016年。arXiv：1607.01759。[30] D. Jurafsky 和 J. H. Martin。《语音与语言处理：自然语言处理、计算语言学与语音识别导论》。第3版草稿。2022年1月12日。[31] K. S. Kalyan、A. Rajasekharan 和 S. Sangeetha。《Ammus：自然语言处理中基于 Transformer 的预训练模型综述》。2021年。arXiv：2108.05542。[32] Y. Kim。《用于句子分类的卷积神经网络》。2014年。arXiv：1408.5882。[33] K. Kowsari、K. Jafari Meimandi、M. Heidarysafa、S. Mendu、L. Barnes 和 D. Brown。《文本分类算法综述》。发表于：Information 10.4（2019年），第150页。[34] A. Krizhevsky、I. Sutskever 和 G. E. Hinton。《使用深度卷积神经网络进行 ImageNet 分类》。发表于：神经信息处理系统进展。2012年，第1097–1105页。[35] T. Kudo 和 J. Richardson。《SentencePiece：一种简单且与语言无关的神经文本处理子词分词器和去分词器》。2018年。arXiv：1808.06226。[36] G. Lample、M. Ballesteros、S. Subramanian、K. Kawakami 和 C. Dyer。《命名实体识别的神经网络架构》。发表于：2016年北美计算语言学协会人类语言技术分会会议论文集。2016年，第260–270页。[37] Y. LeCun 和 Y. Bengio。《用于图像、语音和时间序列的卷积网络》。发表于：脑理论与神经网络手册。3361.10（1995年），第1995页。[38] F. Li、M. Zhang、G. Fu 和 D. Ji。《一种从生物医学文本中抽取实体与关系的神经联合模型》。发表于：BMC生物信息学18.1（2017年），第1–11页。[39] Q. Li 等人。《文本分类综述：从浅层学习到深度学习》。2020年。arXiv：2008.00364。[40] S. Liang 和 R. Srikant。《为什么使用深度神经网络进行函数逼近？》2017年3月3日。arXiv：1610.04161 [cs]。[41] T. Lin、Y. Wang、X. Liu 和 X. Qiu。《Transformer综述》。2021年。arXiv：2106.04554。

[42] M. Lui、J. H. Lau 和 T. Baldwin. “多语言文档的自动检测与语言识别”。载于：计算语言学协会会刊 2（2014），第 27–40 页。[43] C. D. Manning、P. Raghavan 和 H. Schütze。《信息检索导论》。第 39 卷。剑桥大学出版社，剑桥，2008。[44] Y. Meng、J. Huang、G. Wang、C. Zhang、H. Zhuang、L. Kaplan 和 J. Han. “球形文本嵌入”。载于：神经信息处理系统进展 32（2019）。[45] T. Mikolov、K. Chen、G. Corrado 和 J. Dean. “向量空间中词表示的高效估计”。2013。arXiv：1301.3781。[46] T. Mikolov、I. Sutskever、K. Chen、G. S. Corrado 和 J. Dean. “词语和短语的分布式表示及其组合性”。载于：神经信息处理系统进展。2013，第 3111–3119 页。[47] M. Minsky 和 S. A. Papert。《感知器：计算几何导论》。麻省理工学院出版社，1969。[48] K. Nigam、J. Lafferty 和 A. McCallum. “使用最大熵进行文本分类”。载于：IJCAI-99 机器学习与信息过滤研讨会。第 1 卷，第 1 期。瑞典斯德哥尔摩，1999，第 61–67 页。[49] A. van den Oord 等人。“WaveNet：原始音频生成模型”。2016。arXiv：1609.03499。[50] D. W. Otter、J. R. Medina 和 J. K. Kalita. “深度学习在自然语言处理中的应用综述”。载于：IEEE《神经网络与学习系统汇刊》（2020）。[51] J. Pennington、R. Socher 和 C. D. Manning. “GloVe：用于词表示的全局向量”。载于：2014年自然语言处理经验方法会议论文集（EMNLP）。2014，第 1532–1543 页。[52] M. E. Peters、M. Neumann、M. Iyyer、M. Gardner、C. Clark、K. Lee 和 L. Zettlemoyer. “深度上下文化词表示”。载于：北美计算语言学协会人类语言技术分会论文集。2018，第 2227–2237 页。[53] S. Pouyanfar 等人。“深度学习综述：算法、技术与应用”。载于：ACM计算调查（CSUR）51.5（2018），第 1–36 页。[54] X. Qiu、T. Sun、Y. Xu、Y. Shao、N. Dai 和 X. Huang. “自然语言处理预训练模型：综述”。载于：《中国科学：技术科学》63.10（2021年6月23日），第 1872–1897 页。ISSN：1674-7321、1869-1900。DOI：10 . 1007 / s11431 - 020 - 1647 - 3。arXiv：2003.08271。[55] A. Radford、K. Narasimhan、T. Salimans 和 I. Sutskever. “通过生成式预训练提升语言理解能力”。载于：（2018）。[56] F. Reichartz、H. Korte 和 G. Paass. “用于从自然语言文本中抽取关系的依存树核”。载于：欧洲机器学习与知识发现联合会议。施普林格，2009，第 270–285 页。[57] R. Al-Rfou. GitHub上的Cld3。Google，2021年4月8日。网址：https://github.com/google/cld3（访问于2021年4月12日）。[58] M. Sahlgren. “分布假设”。载于：《意大利残障研究期刊》20（2008），第 33–53 页。[59] J. Schmidhuber. “神经网络中的深度学习：综述”。载于：《神经网络》61（2015），第 85–117 页。[60] M. Schuster 和 K. Nakajima. “日语和韩语语音搜索”。载于：2012年IEEE国际声学、语音与信号处理会议（ICASSP）。IEEE，2012，第 5149–5152 页。[61] F. Sebastiani. “自动文本分类中的机器学习”。载于：ACM计算调查（CSUR）34.1（2002），第 1–47 页。[62] Shutterstock。《肮脏、下流、淫秽及其他不良词语列表》。LDNOOBW，2021年4月11日。网址：
h
t
t
p
s://
github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words（访问于2021年4月12日）。[63] X. Song、A. Salcianu、Y. Song、D. Dopson 和 D. Zhou. “快速WordPiece分词”。2021年10月5日。arXiv：2012。15524 [cs]。[64] spaCy。spaCy——工业级自然语言处理。2021。网址：https://spacy.io/。

[65] I. Sutskever、O. Vinyals 和 Q. V. Le. “神经网络序列到序列学习”。载于：神经信息处理系统进展2014，第 3104–3112 页。[66] Y. Tay、M. Dehghani、D. Bahri 和 D. Metzler. “高效Transformer：综述”。2020。arXiv：2009.06732。[67] A. Vaswani 等人。“Attention Is All You Need”。载于：神经信息处理系统进展2017，第 5998–6008 页。[68] L. Wu、A. Fisch、S. Chopra、K. Adams、A. Bordes 和 J. Weston. “Starspace：将万物嵌入其中！”2017。arXiv：1709.03856。[69] Y. Wu 等人。“Google神经机器翻译系统：弥合人工翻译与机器翻译之间的差距”。2016。arXiv：1609.08144。[70] A. Zhang、Z. C. Lipton、M. Li 和 A. J. Smola. “动手学深度学习”。载于：未发表的草稿，检索于19（版本 0.16.1，2021年1月23日），第 1021 页。[71] Y. Zhang、J. Riesa、D. Gillick、A. Bakalov、J. Baldridge 和 D. Weiss. “用于混合语言文本语言识别的快速、紧凑且准确的模型”。2018年10月9日。arXiv：1810.04142 [cs]。[72] J. G. Zilly、R. K. Srivastava、J. Koutnık 和 J. Schmidhuber. “循环高速公路网络”。载于：国际机器学习会议PMLR，2017，第 4189–4198 页。

第2章
预训练语言模型

摘要 本章介绍基于注意力的语言模型的主要架构类型，这些模型用于描述文本中词元的分布：类似 BERT 的自编码器接收输入文本，并为每个词元生成上下文嵌入。类似 GPT 的自回归语言模型接收一个词元子序列作为输入，为每个词元生成上下文嵌入，并预测下一个词元。通过这种方式，文本中的所有词元都可以依次生成。Transformer 编码器-解码器的任务是将一个输入序列翻译成另一个序列，例如用于语言翻译。首先，它们通过自编码器为每个输入词元生成上下文嵌入。然后，这些嵌入作为自回归语言模型的输入，由该模型依次生成输出序列中的词元。这些模型通常在大型通用训练集上进行预训练，并经常针对特定任务进行微调。因此，它们统称为预训练语言模型（PLM）。当这类模型的参数数量变得很大时，通常可以通过提示对其进行指令引导，因此称为基础模型。在后续章节中，我们将介绍训练过程中使用的优化和正则化方法的详细信息。最后，我们将分析模型预测的不确定性，以及如何解释预测结果。

关键词： BERT、语言模型、GPT-2、Transformer、预训练、微调、序列到序列模型：一种计算联合概率或

条件概率的自然语言文本模型称为语言模型，因为它可能涵盖关于语言的所有信息。本章介绍基于注意力的语言模型（语言模型）的主要架构类型。这些模型处理由词元序列构成的文本，即单词、数字等：

- 自编码器 (*AE*)接收输入文本，并为每个词元生成上下文嵌入。这些模型也称为*BERT* 模型，详见第 2.1 节。

- 自回归语言模型 (*AR*)接收输入文本中的一个词元子序列 $v_1, \ldots, v_{t-1}$。它们为每个词元生成上下文嵌入，并用其预测下一个词元 v_t。这样，它们就能依次预测序列中的所有词元。这些模型也称为*GPT*模型，详见第 2.2 节。

- *Transformer* 编码器-解码器的任务是将一个输入序列转换为另一个序列，例如进行语言翻译。首先，它们通过自编码器为每个输入词元生成上下文嵌入。然后，这些嵌入作为自回归语言模型的输入，由该模型依次生成输出序列中的词元。这些模型也称为*Transformer*，将在第 2.3 节中介绍。

本章聚焦于自然语言处理，其中我们考虑文本词元序列。从历史上看，Transformer 编码器-解码器最初由 Vaswani 等人开发， [141] 用于将文本翻译成另一种语言。自编码器 [39]和自回归语言模型 [118] 分别是该 Transformer 编码器-解码器的编码器部分和解码器部分。由于它们在概念上更简单，因此将在前面的章节中介绍。最后一节（第 2.4 节）将介绍在训练期间优化模型、确定模型架构以及估计模型预测不确定性的方法。

事实证明，模型可以先在大型通用文本文件训练集上进行训练，从而掌握正确、流畅语言中词元的分布。随后，可以通过针对小型监督分类任务进行微调等方式，使其适应特定任务。因此，这些模型被称为预训练语言模型。

正如我们稍后将看到的，所有模型都可以应用于任意序列，例如音符、声音、语音、图像，甚至视频。当这些模型的参数数量变得很大时，通常可以通过提示来指示它们，因此被称为基础模型。

2.1 BERT：自注意力与上下文嵌入

常见词语往往有许多不同的含义。例如，对于单词*“bank”*，词汇数据库 WordNet [94] 列出了18种不同的释义，从“倾斜的土地”到“金融机构”。在第1.5节介绍的单词*“bank”* 的简单嵌入中，所有这些含义都被混为一谈。因此，基于这些嵌入对文本进行的解读存在缺陷。

作为一种替代方案，上下文嵌入或上下文化嵌入应运而生。在这种方法中，词嵌入的具体内容取决于单词本身以及特定文档中出现的相邻词语。因此，文本中同一个单词的每次出现，都会根据上下文获得不同的嵌入。自 Transformer [141], 问世以来，人们设计了多种方法来生成这类上下文嵌入，它们通常使用大型文档语料库以无监督方式进行训练。

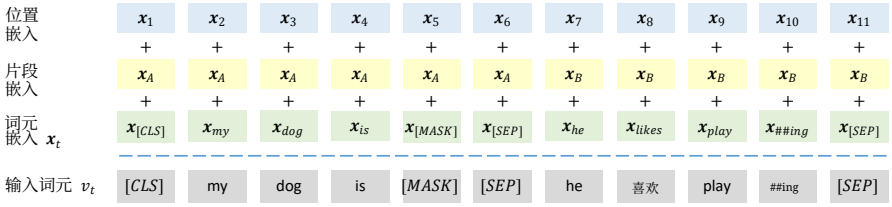


图 2.1 BERT模型的输入由一系列与输入词元相对应的嵌入组成。每个词元都表示为以下各项之和：词元文本的嵌入、其片段指示符的嵌入，以及其位置的嵌入 [39]。

BERT（来自 Transformer 的双向编码器表示）由 Devlin 等人 [39]提出，是生成上下文嵌入最重要的方法。BERT 基于注意力的概念 [8]以及 Vaswani 等人 [141]的早期研究。**注意力**这一概念的灵感来源于大脑的一种机制：在处理大量信息时，大脑往往会聚焦于记忆中具有显著特征的部分。Rush [126]解释了计算细节。

2.1.1 BERT输入嵌入与自注意力

BERT 的输入是一些文本，这些文本会被转换为词元，例如使用具有指定词汇表大小（如 30,000）的 WordPiece 分词器（第 1.2 节）进行转换。这意味着，像“狗”这样的常见单词会由专属词元表示，而像“*playing*”这样的较罕见单词则会拆分为多个词元，例如“*play*”和“*##ing*”，其中“*##*”表示该词元是某个单词的一部分。由于所有字符都会保留为词元，因此任意单词都可以用少量词元表示。此外，还有一些特殊词元，例如位于输入文本首个位置的 [*CLS*]，以及用于标记文本片段结尾的两个“[*SEP*]”词元。最后，在训练期间还会用到 [*MASK*]词元，后文将对此进行说明。每个词元都由一个词元嵌入表示，即一个长度固定的向量 d_{emb}，例如 $d_{emb} = 768$。长度不一的输入序列会通过特殊的填充词元补齐至最大长度。

由于所有词元嵌入都是同时处理的，因此需要标明各词元在输入文本中的位置。因此，每个位置都用位置嵌入 嵌入$_1, \ldots,$嵌入$_T$标记，其长度与词元嵌入相同，用于编码位置索引。BERT 论文通过可训练的嵌入来编码位置编号，并将其加到输入词元嵌入 [39]中。最后，BERT 会比较第一个和第二个输入片段。因此，算法需要知道每个词元属于第一个还是第二个片段。这一点也通过可训练的片段嵌入进行编码，并将其加到词元嵌入和位置嵌入上。所有嵌入的总和用作 BERT 的输入嵌入 。示例如图 2.1 所示。

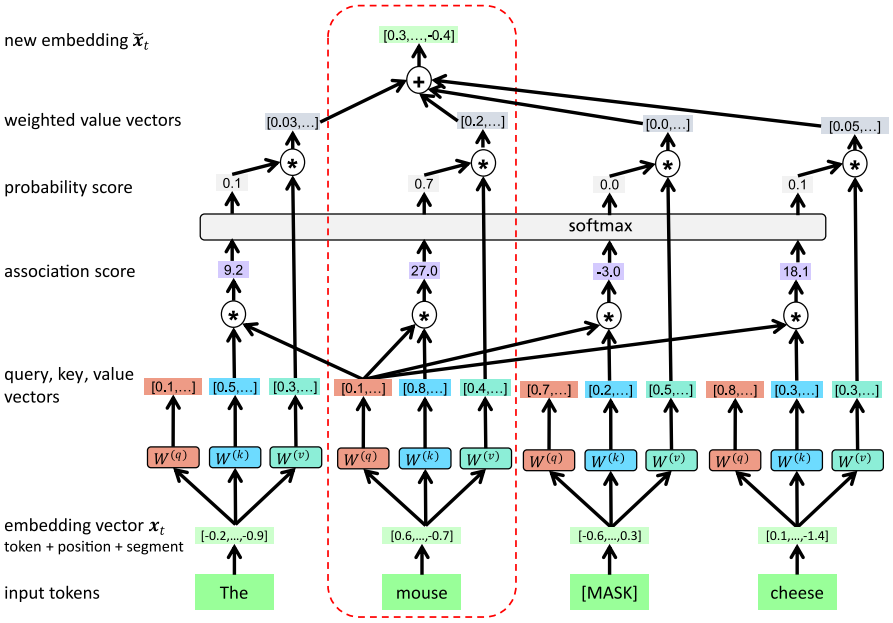


图 2.2 通过自注意力计算单个词元 *"mouse"* 的上下文嵌入。加入"奶酪"的嵌入后，"mouse"的嵌入可以向"啮齿动物"的意义偏移，而远离"计算机指点设备"的意义。输入序列中的每个单词都会计算出这样的嵌入。

通过自注意力生成上下文嵌入

BERT 首先为输入序列 $v_1, \ldots, v_T$ 中的每个词元 v_t 提供长度为 d_{emb} 的输入嵌入 $\boldsymbol{x}_t$。这些嵌入经线性映射转换为所谓的查询向量 $\boldsymbol{q}_t$、键向量 $\boldsymbol{k}_t$ 和值向量 $\boldsymbol{v}_t$。它们通过将 $\boldsymbol{x}_t$ 与矩阵 $\boldsymbol{W}^{(q)}$、$\boldsymbol{W}^{(k)}$ 和 $\boldsymbol{W}^{(v)}$ 相乘计算得出，其维度分别为 $d_{emb} \times d_q$、$d_{emb} \times d_q$ 和 $d_{emb} \times d_v$

$$\boldsymbol{q}_t^\intercal = \boldsymbol{x}_t^\intercal \boldsymbol{W}^{(q)} \qquad \boldsymbol{k}_t^\intercal = \boldsymbol{x}_t^\intercal \boldsymbol{W}^{(k)} \qquad \boldsymbol{v}_t^\intercal = \boldsymbol{x}_t^\intercal \boldsymbol{W}^{(v)}. \tag{2.1}$$

请注意，查询向量和键向量的长度相同。接着，计算目标词元的查询向量与序列中所有词元的键向量之间的标量积$\boldsymbol{q}_r \boldsymbol{k}_t$ $\boldsymbol{q}_r$ v_r $\boldsymbol{k}_t$：

$$(\alpha_{r,1}, \ldots, \alpha_{r,T}) = \text{softmax}\left(\frac{\boldsymbol{q}_r^\intercal \boldsymbol{k}_1}{\sqrt{d_k}}, \ldots, \frac{\boldsymbol{q}_r^\intercal \boldsymbol{k}_T}{\sqrt{d_k}}\right). \tag{2.2}$$

每个标量积都会得到词元之间的实值关联分数$(\boldsymbol{q}_r \boldsymbol{k}_t)/\sqrt{d_k}$，该分数取决于矩阵 $\boldsymbol{W}^{(q)}$ 和 $\boldsymbol{W}^{(k)}$。该关联分数被

称为缩放点积注意力。它通过 softmax 函数归一化为概率得分 $\alpha_{r,\ t}$。因子 $1/\sqrt{d_k}$ 可避免数值过大，否则 softmax 函数的梯度会非常小。利用这些权重，对所有序列元素的值向量 $\boldsymbol{v}_t$ 进行加权平均，从而得到目标词元 v_r 的新嵌入 $\check{\boldsymbol{x}}_r$，其长度为 d_v：

$$\check{\boldsymbol{x}}_r = \alpha_{r,1} * \boldsymbol{v}_1 + \cdots + \alpha_{r,T} * \boldsymbol{v}_T . \tag{2.3}$$

这种算法称为**自注意力**，最早由 Vaswani 等人提出 [141]。图 2.2 展示了对第 r 个词元“老鼠”进行计算的过程。请注意，得到的嵌入是上下文嵌入，因为它包含输入文本中所有词语的信息。每当标量积 $\boldsymbol{q}_r \boldsymbol{k}_t$ 较大时，$\boldsymbol{v}_t$ 的一个分量就会获得较高的权重。它衡量 $\boldsymbol{x}_r$ 与 $\boldsymbol{x}_t$ 之间特定形式的相关性；当向量 $\boldsymbol{x}_r \boldsymbol{W}^{(q)}$ 与 $\boldsymbol{x}_t \boldsymbol{W}^{(k)}$ 方向相同时，该相关性达到最大。

一般而言，自注意力机制并不对称，因为矩阵 $\boldsymbol{W}^{(q)}$ 和 $\boldsymbol{W}^{(k)}$ 并不相同。如果词元 v_i 对词元 v_j 有很高的注意力（即 $\boldsymbol{q}_i \boldsymbol{k}_j^\intercal$ 很大），这并不一定意味着 v_j 也会高度关注词元 v_i（即 $\boldsymbol{q}_j \boldsymbol{k}_i^\intercal$ 也很大）。因此，v_i 对 v_j 的上下文嵌入的影响，与 v_j 对 v_i 的上下文嵌入的影响不同。考虑以下示例文本“弗雷德把玫瑰送给了玛丽”。这里，单词“送给”与其余单词之间存在不同的关系。“弗雷德”是实施赠送行为的人，“玫瑰”是被赠送的物体，而“苏珊”是这些物体的接受者。显然，这些语义角色关系并不对称。因此，可以通过不同的矩阵 $\boldsymbol{W}^{(q)}$ 和 $\boldsymbol{W}^{(k)}$ 来捕捉这些关系，并将其编码到嵌入中。

自注意力能够缩短计算路径，并提供直接比较输入序列中相距较远元素的途径，例如句子中的代词及其先行词。注意力中的乘法交互能够根据当前输入动态调整计算方式，相较于多层感知机和卷积神经网络僵化的固定权重计算，提供了更灵活的替代方案。这对于语言建模尤其有用。例如，在处理句子“她用 X” 吃了冰淇淋。时，前馈网络总是以相同的方式处理句子，而基于注意力的模型则可以根据输入调整计算方式：如果 X 是“勺子”，就更新单词“吃”的上下文嵌入；如果 X 指的是“草莓” [17]，则更新“冰淇淋”的嵌入。

实际上，所有查询向量、键向量和值向量都由 $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}^{(q)}$、$\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}^{(k)}$、$\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}^{(v)}$ 并行计算，其中 $\boldsymbol{X}$ 是输入嵌入 [141] 的 $T \times d_{emb}$ 矩阵。查询向量 $\boldsymbol{q}_t$、键向量 $\boldsymbol{k}_t$ 和值向量 $\boldsymbol{v}_t$ 分别是 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$ 的各行。然后，通过一个大型矩阵表达式计算自注意力输出矩阵

$$\check{\boldsymbol{X}} = \textsc{Attl}(\boldsymbol{X}) = \textsc{Attl}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\intercal}{\sqrt{d_k}}\right)\boldsymbol{V}, \tag{2.4}$$

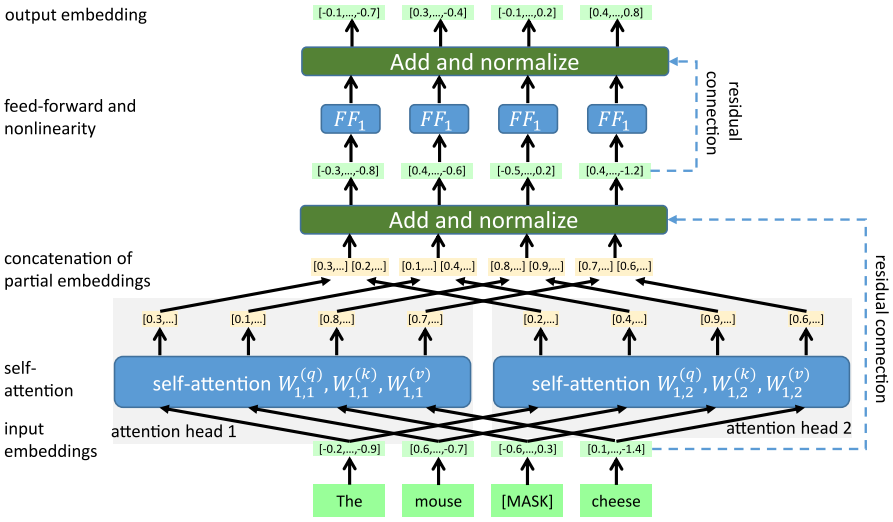


图 2.3 多头自注意力使用不同的矩阵 $\boldsymbol{W}_{(q)l,\ m}$，$\boldsymbol{W}_{(k)l,\ m}$，以及 $\boldsymbol{W}_{(v)l,\ m}$ 为每一层 l 和每个注意力头 m 计算自注意力。这样便可计算词元对（例如“老鼠”和“奶酪”）之间关联的不同方面。随后，将得到的嵌入拼接起来，并通过前馈网络进行变换。此外，残差连接和层归一化有助于提升训练收敛速度 [39]。

˘从而得到一个 $T \times d_v$ 矩阵 $\boldsymbol{X}$。其第 r 行包含第 r 个词元 v_r 的新嵌入 $\check{\boldsymbol{x}}_r$。

除了缩放点积注意力 2.2 之外，人们还提出了许多其他兼容性度量。不过，正如综述 [27, 46] 中所述，这些方法在预训练语言模型中很少使用。

事实证明，单个自注意力模块不足以表征这些词元。因此，在一层中 d_{head}，使用不同的矩阵 $\boldsymbol{W}_m^{(q)}, \boldsymbol{W}_m^{(k)}$ 和 $\boldsymbol{W}_m^{(v)}, m = 1、\ldots\ldots、d_{\text{head}}$ 并行计算自注意力，从而得到部分新的嵌入

$$\check{\boldsymbol{X}}_m = \text{Attl}(\boldsymbol{X}\boldsymbol{W}_m^{(q)}, \boldsymbol{X}\boldsymbol{W}_m^{(k)}, \boldsymbol{X}\boldsymbol{W}_m^{(v)}). \tag{2.5}$$

新兴的部分嵌入 $\check{\boldsymbol{x}}_{m,\ t}$ 能够专注于互补的语义方面，这些方面在训练过程中逐渐形成。

$\text{BERT}_{\text{BASE}}$ 模型具有 $d_{\text{head}}=12$ 个这样的并行注意力头。这些注意力头嵌入的长度仅为原始长度 d_{emb} 的 d_{emb}/d_{head}。所得嵌入会拼接起来，并与一个 $(d_{\text{head}} * d_v) \times d_{emb}$- 矩阵 $W^{(o)}$ 相乘，从而得到中间嵌入矩阵

$$\check{\boldsymbol{X}} = \left[\check{\boldsymbol{X}}_1, \ldots, \check{\boldsymbol{X}}_{d_{\text{head}}}\right] \boldsymbol{W}_0, \tag{2.6}$$

其中 $\boldsymbol{W}_0$ 是一个参数矩阵。如果输入嵌入的长度为 d_{emb}，则查询、键和值向量的长度设为 $d_k = d_v = d_{emb}/d_{\text{head}}$。因此，拼接后会再次生成一个 $T \times d_{emb}$ 矩阵 $\boldsymbol{X}$。这种设置称为多头自注意力。由于各个注意力头的维度有所降低，总计算成本与采用完整维度的单头注意力相近。

随后，$\boldsymbol{X}$ 中的每一行，即中间嵌入向量 $\check{\boldsymbol{x}}_t$，都会经过一个全连接层 Fcl 和 ReLU 激活，再进行一次线性变换 [141]

$$\tilde{\boldsymbol{x}}_t^\intercal = \text{FCL}(\check{\boldsymbol{x}}_t) = ReLU(\check{\boldsymbol{x}}_t^\intercal * \boldsymbol{W}_1 + \boldsymbol{b}_1^\intercal) * \boldsymbol{W}_2 + \boldsymbol{b}_2^\intercal. \tag{2.7}$$

矩阵 $\boldsymbol{W}_0$，$\boldsymbol{W}_1$，$\boldsymbol{W}_2$ 以及向量 $\boldsymbol{b}_1$，$\boldsymbol{b}_2$ 都是参数。对于序列中的每个词元 v_t，这些变换都相同，从而得到嵌入 $\tilde{\boldsymbol{x}}_t$。

为提高训练速度，我们添加残差连接 作为“旁路”，直接复制输入。研究表明，这对多层图像分类器的优化 [54]非常有帮助。此外，使用层归一化 [6] 进行正则化（第 2.4.2 节），如图 2.3 所示。多头自注意力（2.5）、拼接（2.6）和全连接层（2.7）共同构成一个编码器块。

这一过程会针对多个 k 层重复进行，每层使用不同的编码器块，并将一个块的输出嵌入用作下一块的输入嵌入。图 2.4 展示了这种架构。最后一个编码器块的嵌入 $\tilde{\boldsymbol{x}}_{k,\,t}$ 包含所需的上下文嵌入。编码器块的结构克服了循环神经网络的局限（即循环神经网络的顺序特性），使输入序列中的每个词元都能直接与序列中的其他所有词元建立关联。$\text{BERT}_{\text{BASE}}$ 有 $k=12$ 个编码器块。它由 Google 的 Devlin 等人 [39]开发。关于自注意力实现方式的更多详细信息，请参阅这些论文 [38, 41, 126]。

2.1.2 通过预测掩码词元训练 BERT

BERT模型有大量未知参数。这些参数通过两步流程进行训练。

- 预训练 使模型能够以无监督的方式获取有关语言的一般知识。模型的任务是补全文本中缺失的词语。由于不需要人工标注，预训练可以使用大型文本语料库。
- 微调会针对特定任务调整预训练模型，例如情感分析。在此过程中，利用规模较小的带标签训练数据集调整模型参数，以解决该任务。

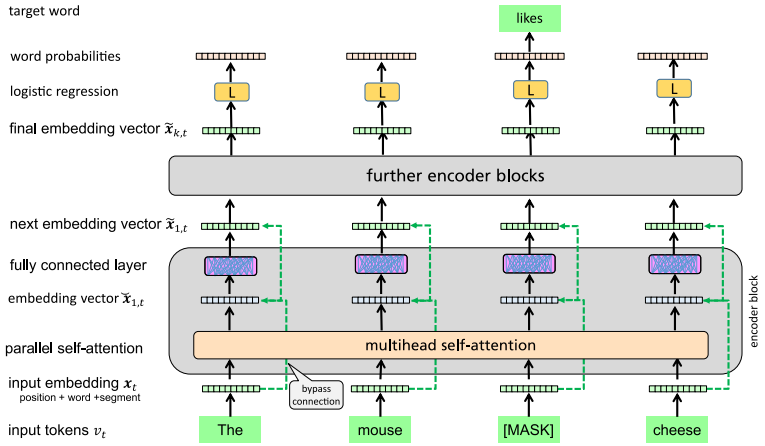


图 2.4 BERT 在每个编码器块中并行计算上下文嵌入。一个编码器块的输出嵌入用作下一个编码器块的输入嵌入。最后，逻辑分类器 L 使用最后一个编码器块的相应上下文嵌入作为输入，预测掩码词元。

由于模型可以通过迁移学习利用预训练期间获得的知识，因此在微调任务上的性能比不进行预训练时好得多。

为了预训练模型参数，设计了一项训练任务：掩码语言模型 （*MLM*）。训练文档中约 15% 的输入词元会被选中进行预测，预测由逻辑分类器完成（第 1.3 节）

$$p(V_t|v_1,\ldots,v_{t-1},v_{t+1}\ldots,v_T) = \text{softmax}(A\tilde{\boldsymbol{x}}_{k,t} + \boldsymbol{b}), \tag{2.8}$$

接收最后一层在位置 t 处的嵌入 $\tilde{\boldsymbol{x}}_{k,\ t}$ 作为输入，以预测位置 t处可能词元的随机变量 V_t 。这种方法避免了单词间接“看到自身”的循环。

要预测的词元必须进行更改，否则预测会变得轻而易举。因此，被选中用于预测的词元将被替换为:

- 在 80% 的时间里替换为一个特殊的 [*MASK*]词元（例如，*“the mouse likes cheese”*会变成*“the mouse* [*MASK*]*cheese”*）；
- 在 10% 的时间里使用一个随机词元（例如，*“themouse likescheese”* 变为*“themouse absent cheese”*）；
- 在 10% 的时间里使用未更改的标签词元（例如，*“the mouse likes cheese”*变为 *“themouse likes cheese”*）。

引入了第二种和第三种变体，因为预训练与后续微调之间存在差异，微调时没有 [掩码] 词元。作者通过偶尔将 [掩码] 替换为原始词元，或从词汇表中采样来缓解这一问题。请注意，在 1.5% 的情况下，会插入一个随机

词元。这种偶尔加入的噪声促使 BERT 在双向上下文编码中减少对掩码词元的偏见（尤其是在标签词元保持不变时）。为了预测掩码词元，BERT 必须将有关该词元的所有知识集中到最后一层对应的输出嵌入中，而该嵌入是逻辑分类器的输入。因此，它通常被称为**自编码器**，能够生成极其丰富的输出嵌入。

除了预测掩码词元外，BERT 还必须预测下一句是随机选取的句子，还是实际的后续句子（**下一句预测**）。因此，BERT 必须考虑文本中两个连续部分之间的关系。分类时仍使用一个逻辑分类器，该分类器接收第一个 [*CLS*] 词元的嵌入。然而，这项任务对 BERT 的性能并没有重大影响，因为 BERT 只是学会了判断两个句子的主题是否相似 [158]。

在图 2.4 中，任务是针对输入文本 “这只老鼠 [*MASK*]奶酪”，预测词元 “喜欢” 具有较高的概率。在训练开始时，这一概率会非常小（≈ 1/词元数量）。通过反向传播，可以确定每个未知参数的导数，从而指示应如何调整参数以提高 “喜欢”的概率。BERT 的未知参数包括词汇表中每个词元的输入嵌入、每个位置的位置嵌入、每一层 l和每个注意力头 m(2.4)对应的矩阵$\boldsymbol{W}_{l\,m}^{(q)}$, $\boldsymbol{W}_{(k)l,m}$, $\boldsymbol{W}_{l\,m}^{(v)}$，全连接层的参数（2.7），以及逻辑分类器的 $A, \boldsymbol{b}$（2.8）。BERT 使用 Adam 算法 [69] 进行随机梯度下降。

$\text{BERT}_{\text{BASE}}$ 模型的隐藏层维度为 $d_{emb}=768$，包含 $k=12$ 个编码器块，每个编码器块具有 $d_{\text{head}}=12$ 个注意力头，参数总数为 1.1 亿。$\text{BERT}_{\text{LARGE}}$ 模型的隐藏层维度为 $d_{emb}=1024$，包含 $k=24$ 个编码器块，每个编码器块具有 $d_{\text{head}}=16$ 个注意力头，参数总数为 3.4 亿 [39]。英文维基百科和一个包含 33 亿词语的书籍语料库经由 WordPiece 分词器 [154]进行编码，该分词器的词汇表包含 30,000 个词元，并被用于预训练 BERT。无需人工对文本进行标注，因此该训练是自监督的。预训练在 64 个 TPU 芯片上耗时 4 天。微调可在单个 GPU 上完成。

为了预测掩码词元，模型必须学习多种类型的语言理解特征：句法（[掩码] 是放置动词的合适位置）、语义（例如，老鼠更喜欢奶酪）、语用、共指等。请注意，输入序列中每个词元的计算都可以并行处理，从而消除了循环神经网络中的序列依赖性。这种并行使 BERT 及相关模型能够充分利用现代单指令多数据（SIMD）硬件加速器（如 GPU/TPU）的全部能力，进而促进在规模空前的数据集上训练自然语言处理模型。在句子中重建缺失词元长期以来一直被用于心理学。因此，根据格式塔理论（一种心理学流派）中的“闭合”概念，预测掩码词元也被称为完形填空任务。

结果表明，BERT 在掩码词元预测方面表现出色，增加编码器块还能显著提高准确率。例如，BERT 能以 45.9% 的准确率预测原始单词（或单词的一部分），尽管在许多情况下，目标位置 [125] 都有多个有效值。

与传统语言模型不同，MLM 会考虑掩码目标词元之前和之后的词元。因此，它被称为 双向编码器。此外，自注意力无需应用循环模型，就能直接建立与远距离词元的关系。最后，自注意力的速度很快，因为它可以针对编码器块的所有输入词元并行计算。

2.1.3 将 BERT 微调至下游任务

神经网络早在许多年前就已经进行预训练了 [16],，但近年来预训练的成功变得更加显著。在预训练期间，BERT 学习了语言的一般句法和语义属性。这些知识可以在后续的特殊训练任务中加以利用，即使用经过修改的训练任务进行微调。这种方法也称为迁移学习，因为在预训练期间获得的知识会被转移到相关应用中。与其他模型相比，BERT 只需进行最少的架构更改，就能适用于广泛的自然语言处理任务。在发布时，BERT 在各种自然语言处理任务上提升了最先进水平。

通常，微调任务需要进行分类，这可以通过将逻辑分类器 L 应用于 BERT 最后一个编码器块中位于位置 1 的 [*CLS*] 词元的输出嵌入 $\tilde{\boldsymbol{x}}_{k,\,1}$ 来完成。微调任务有不同的类型，如图 2.5 所示。

- 文本分类将一个句子划分为两个或多个类别。例如，将餐厅评论分为正面或负面，或评估句子是否属于优质或糟糕的英语。此处将起始词元的输出嵌入[*CLS*] 作为 L 的输入，以生成最终分类结果。
- 文本对分类比较由“[*SEP*]”分隔的两个句子。示例包括判断第二个句子是蕴含、矛盾于还是中性于第一个句子，或判断两个句子是否语义等价。同样，将起始词元的输出嵌入 [*CLS*] 作为 L的输入。有时会将多个句子与根句子进行比较。此时，会针对每一对句子计算输出，并联合归一化为概率。

- 词语标注为输入文本中的每个单词或词元标记特定属性。例如，命名实体识别（*NER*）会为词元标注五种名称类别（例如“人”、“位置”、. . .,“其他”）。此处，对每个词元位置 t的输出嵌入$\tilde{\boldsymbol{x}}_{k,t}$ 应用相同的逻辑回归模型 L，并生成不同实体类别的概率向量。
- 片段预测标签表示文本中的一段较短词元序列。问答就是一个示例。BERT 的输入由一个问题和“[*SEP*]” 后接一段上下文文本组成，并假设其中包含答案。在这里，两个不同的逻辑分类器分别应用于上下文中每个词元的输出嵌入，$_{,\,t}$ 并生成答案在何处开始的概率

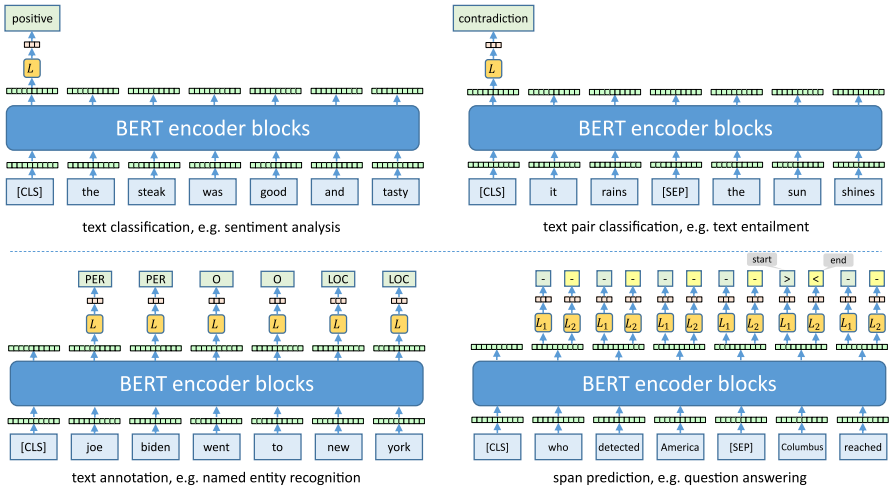


图 2.5 微调时，BERT会增加一个额外层，其中包含一个或多个逻辑分类器 L，并将最后一层的嵌入作为输入。这种设置可用于文本分类，以及将文本与 $[CLS]$ 的嵌入进行比较，并将其作为逻辑分类器的输入。对于序列标注，L 会为序列中的每个词元预测一个类别。对于片段预测，两个逻辑分类器 L_1 和 L_2 分别预测回答短语 [39]的起始位置和结束位置。

以及在特定位置结束的概率。选择起始得分与结束得分之和最高的有效片段（即结束位置不早于起始位置）作为答案。示例输入为“$[CLS]$ 凯撒何时去世？ $[SEP]$ ……*3*月十五日，公元前*44*年，凯撒遭一群反叛元老刺杀…… ”，其中问题的答案是片段“三月十五日$_{start}$，公元前*44*年$_{end}$”。片段预测可用于多种类似任务。

因此，BERT在微调时只需增加一个包含一个或多个逻辑分类器的额外层。在下游应用的微调过程中，逻辑模型的参数从头开始学习，通常也会调整预训练BERT模型的所有参数。掩码语言模型和下一句预测的逻辑分类器参数不会在微调时使用。

2.1.4 注意力与嵌入的可视化

根据 Bengio 等人 [14], 的观点，良好的语言表示应捕捉文本数据中蕴含的语言规则和常识知识，例如词汇意义、句法关系、语义角色以及语言使用的语用学。BERT 的上下文词嵌入可视为朝这一方向迈出的一大步

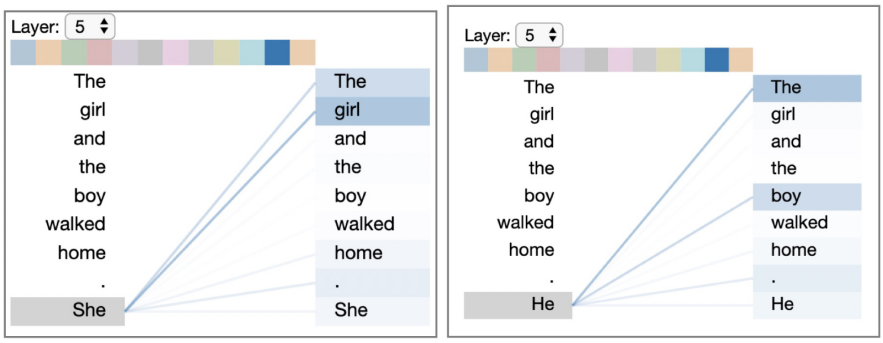


图 2.6 使用 BertViz [142]对 BERT 模型第五层中的特定自注意力进行可视化。如果下一句中包含代词“她”，它就会与“那个女孩”相关联。如果将这个代词改为“他”，它就会与“那个男孩”相关联。图像由作者使用 BertViz [142], 创建，已获作者许可。

。它们可用于消歧同一个单词的不同意义。

BERT 的自注意力机制会计算词元之间大量的“关联”，并根据这些关联的强度合并嵌入。如果 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ 是输入词元 $v_1, \ldots, v_T$ 的嵌入，那么查询 $\boldsymbol{q}_r = \boldsymbol{x}_r \boldsymbol{W}^{(q)}$ 与键向量 $\boldsymbol{k}_t = \boldsymbol{x}_t \boldsymbol{W}^{(k)}$ 之间的关联 $\boldsymbol{q}_r \boldsymbol{k}_t^\intercal$ 由式（2.1）确定。然后，将值向量 $\boldsymbol{v}_t = \boldsymbol{x}_t \boldsymbol{W}^{(v)}$ 按归一化后的关联加权求和，从而得到新的嵌入（2.3）。

这一过程会使用不同的矩阵重复进行 $\boldsymbol{W}_{l\,m}^{(q)}$，$\boldsymbol{W}_{l\,m}^{(k)}$，$\boldsymbol{W}_{l\,m}^{(v)}$ 分别应用于 m 自注意力头和 l 层。因此，每一层和每个注意力头中的新嵌入都会捕捉各层嵌入之间关系的不同方面。对于 $\text{BERT}_{\text{BASE}}$，我们有 $l = 12$ 层，并且每层有 $m = 12$ 个双向自注意力头，从而产生 144 种不同的“关联”或自注意力。对于输入句子 *“The girl andthe boy went home. She entered the door.”*，图 2.6 左侧显示了 144 个自注意力头中的一个所对应的关联强度。句子中每一对词元之间都会计算一个注意力值，其强度通过不同宽度的线条表示。我们可以看到，代词 *“she”* 与 *“thegirl”* 存在很强的关联。在后续计算中（参见图 2.2），通过将其嵌入与 *“the”* 和 *“girl”* 的嵌入合并，单词 *“she”* 得以消歧，从而生成一个新的 上下文嵌入，其中包含它与 *“girl”*之间的关系。在图的右侧，对输入 *“The girl and the boy went home. He entered thedoor.”* 进行处理。随后，模型将 *“boy”* 与*“he”*建立关联。

图 2.7 展示了句子 *“[CLS]thecat saton the mat [SEP] thecat lay on the rug [SEP]”*中的部分自注意力模式。这些自注意力模式会自动优化，使它们共同实现对掩码词元的最优预测。可以看到，特殊词元*[CLS]*和 *[SEP]* 常常是注意力的显著目标。它们通常充当

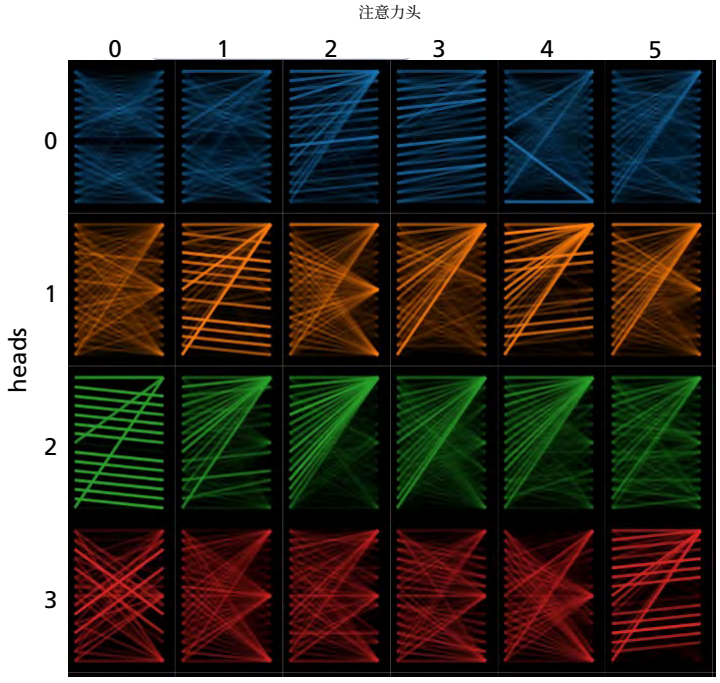


图 2.7 使用 BertViz 对句子 “[*CLS*] 猫坐在垫子上 [*SEP*] 猫躺在地毯上[*SEP*]” 计算出的 144 个自注意力模式中的一部分的可视化。经作者 [142] 慷慨许可后转载图像。

整个句子 [124]的代表。不过，在多层预训练语言模型中，不同注意力头生成的嵌入会被拼接并经过非线性变换。因此，单个注意力头的注意力模式并不包含完整信息 [124]。每当矩阵采用随机初始化时，如果使用新的随机参数值重新开始训练，自注意力模式就会完全不同。不过，词元之间注意力的整体模式仍会相似。

图 2.10 左侧展示了 *“bank”* 词元嵌入的六种不同词义在 *Senseval-3* 数据集 中的分布图，这些嵌入由 *T-SNE* [140]投影到二维空间。不同词义由不同颜色标识，并形成各自明显分离的簇。难以区分的词义，例如“银行大楼”和“金融机构”，显示出明显的重叠 [153]。该图表明，BERT 嵌入能够区分出现频率足够高的单词的不同词义。

关于自注意力内部工作机制的讨论仍在进行中。Tay 等人 [134] 通过实证评估了点积 $\boldsymbol{q}_r \boldsymbol{k}_s$ 在自然语言处理任务中的重要性，并得出结论：查询键交互“有用，但并没有那么重要”。因此，他们推导出了其他公式，其中一些

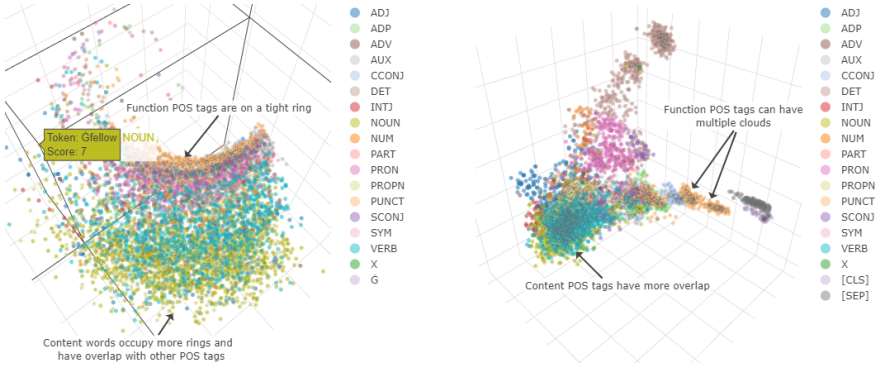


图 2.8 使用 PCA 对嵌入进行可视化，并显示相应的词性标签。左侧是 GPT-2 第 0 层中位置为 > 0 的词元嵌入；不同词性标签的词元形成带状结构，其中虚词靠近顶部。右侧显示的是 BERT 第 0 层的嵌入。经作者 [66]友好授权转载图片。

在某些情况下表现良好，但在其他情况下则失败。de Santana Correia 等人 [37]对注意力方法进行了综述。存在多种不同的注意力机制，用于计算嵌入向量之间的关联 [50, 61, 104, 151]。然而，目前大多数大规模模型仍然采用原始的缩放点积注意力，仅进行一些细微变体，例如使用其他激活函数和正则化器（参见第 3.1.4 节）。

(2.7) 中的全连接层 $\mathrm{Fcl}(\check{x}_l)$ 包含 BERT 2/3 的参数，但其在网络中的作用几乎尚未得到讨论。Geva 等人 [49]表明，全连接层充当键值记忆，其中每个键都与训练样本中的文本模式相关联，而每个值都会在输出词汇表上诱导出一个分布。对于某个键，作者会检索使该键产生最高激活的训练输入。专家能够为每个键赋予一种或多种解释结果。通常，较低的全连接层与浅层模式相关联，这些模式往往共享最后一个单词。较高层则以更多语义模式为特征，用于描述相似的上下文。作者证明，前馈层的输出是其记忆的组合。

2.1.5 BERT 的自然语言理解

预训练语言模型的一个重要目标是自然语言理解（自然语言理解）。这无法通过单一任务进行评估，而是需要涵盖不同领域的一系列基准测试，以评估机器理解自然

表格 2.1 GLUE语言理解任务。$\text{BERT}_{\text{LARGE}}$ 在微调数据集 [38] 上训练了三个训练轮次。最终模型的性能列于最后一列，平均值为82.1。

Task	描述	示例	指标	BERT
CoLA	句子是否符合语法还是不合语法?	*"这栋建筑比那栋。"* → 不合语法	Matthews相关性	60.5
SST-2	这部电影是正面、负面，还是中性?	*"这部电影幽默、聪明、在视觉上富有创意，而且最重要的是，充满活力。"* → 正面	准确率	94.9
MRPC	这个句子是 *B* a句子的释义 *A*?	*A*：*"今天，台湾报告新增 35 例感染病例。"* *B*：*"台湾宣布新增 35 例疑似病例，时间是中午。"* → 释义	准确率	89.3
STS-B	句子有多相似 *A* 以及 *B*?	*A*：*"大象正沿着小径行走。"* *B*：*"一群大象正沿着小径行走。"* → 相似	皮尔逊 / 斯皮尔曼相关系数	86.5
QQP	这两个问题是否相似?	*A*：*"使用 VPN 时，如何提高互联网连接速度？"* *B*：*"如何通过 DNS 黑客手段提高互联网速度？"* → 不相似	准确率	72.1
MNLI-mm	句子 *A*是蕴含还是与句子 *B*矛盾?	*A*：*"旅游信息办公室会很有帮助。"* *B*：*"旅游信息办公室从来都毫无帮助。"* → 矛盾	准确率	85.9
QNLI	句子 *B* 是否包含问题的答案句子 *A*?	*A*：*"有时被认为是维吉尔作品的小诗集是什么？"* *B*：*"若干首小诗汇集成了《维吉尔附录》，人们常认为这些诗歌是他创作的。"* → 包含回答	准确率	92.7
RTE	句子 *A*是否蕴含句子 *B*?	*A*：*"尤努斯发起了小额信贷革命，为5万名乞丐提供资金，格莱珉银行恭敬地称他们为'奋斗成员。'"* *B*：*"尤努斯支持了超过50,000名奋斗成员。"* → 蕴含	准确率	70.1
WNLI	句子 *B* 替换句子 *A*的代词，改为名词——这是正确的名词吗?	*A*：*"莉莉和唐娜交谈，打断了她的注意力。"* *B*：*"莉莉和唐娜交谈，打断了莉莉的注意力。"* →错误	准确率	60.5

语言文本并获取语言、常识和世界知识的能力。因此，预训练语言模型会针对相应的现实世界下游任务进行微调。

GLUE [146] 是自然语言理解领域一项重要的基准测试。它由九项自然语言理解任务组成，提供公开的训练数据，并设有使用私有测试数据的评测服务器。其基准测试涵盖多个不同方面，这些方面可以表述为分类问题：

- 判断句子的情感是正面还是负面（SST-2）。
- 将句子归类为语法上可接受或不可接受（CoLA）。
- 检查两个句子是否相似或互为释义（MPRC、STS-B、QQP）。
- 判断第一个句子是否蕴含第二个句子（MNLI、RTE）。
- 检查句子 *B* 是否包含问题 *A* 的答案（QNLI）。
- 从一组选项中指出代词所指的对象（WNLI）。

每项任务都可以表述为文本分类或文本对分类问题。模型的性能以一个平均值概括；人工标注员的这一数值为 87.1 [145]。通常会有一个在线排行榜，记录不同模型的性能。PapersWithCode 网站上有一个庞大的排行榜资源库。

表 2.1 通过示例描述了这些任务，并报告了 $\text{BERT}_{\text{LARGE}}$ 的性能。BERT 将平均准确率的最先进水平从 75.2 提升到了 82.1%。这是一个显著的提升，但该数值仍远低于人类表现 87.1，仍有很大的提升空间。第 4.1 节介绍了近期自然语言理解的基准测试结果，重点讨论要求更高的 SuperGLUE 及其他基准测试。

BERT 在其他微调任务上的性能

预训练数据足以调整大量 BERT 参数，并学习语言的诸多细致特征。预训练的训练数据量通常远高于微调所使用的数据量。微调通常只需要在微调训练数据上进行两到三轮训练。因此，随机梯度优化器只会略微改变大多数参数，使其相对接近最优的预训练参数。 consequently，模型通常能够保留其关于一般语言的信息，并将这些信息与微调任务的信息结合起来。

由于 BERT 可以重新利用其在预训练期间获得的语言一般知识，即使微调训练数据很少，它也能取得出色的结果[39]。

- **CoNLL 2003** [128]是一个用于命名实体识别 （*NER*）的基准数据集，其中每个词元都必须标注命名实体标签，例如 PER（人物实体）、LOC（位置）、.....、O（无姓名）（第 5.3 节）。该任务涉及文本标注，需要为每个输入词元预测一个标签。BERT 将测试数据上的最先进水平从 92.6% 提高到了 92.8% 的 F1值。
- **SQuAD 1.0** [120] 是一个由 10 万组三元组组成的集合，包含问题、上下文和答案。此任务要求在上下文中标出答案词元所在的跨度。例如，问题“奥古斯都何时去世？”的答案是“公元 *14* 年”，需要在上下文“.....奥古斯都于公元 *14* 年去世.....”中标出（第 6.2 节）。采用片段预测后，BERT 将 SQuAD 的最先进水平从 91.7% 提高到了 93.2%，而人类表现测得为 91.2%。

这些实验积累了大量证据，揭示了 BERT 的优势和劣势 [124]。相关讨论见第 4.2 节。

总之，BERT模型的出现标志着自然语言处理领域开启了一个新时代。它结合了两项预训练任务，即预测掩码词元以及判断第二个句子是否与第一个句子匹配。通过无监督预训练和有监督微调进行迁移学习成为新的标准。

2.1.6 计算复杂度

说明训练预训练语言模型所需的计算开销很有帮助。这决定了训练更大模型所需的时间，而更大的模型可以大幅提升语言表示的质量。假设 D 是隐藏嵌入的大小，且输入序列的长度为 T，那么全连接层 Fcl 的中间维度设为 $4D$，键和值的维度按照 Vaswani 等人 [141] 的设定为 D/H。然后，根据 Lin 等人 [81]，我们可以得到自注意力和逐位置 Fcl （ 2.7 ）的以下计算复杂度和参数数量：

模块	复杂度	# 参数
自注意力	$O(T^2 * D)$	$4D^2$
逐位置全连接层	$O(T * D^2)$	$8D^2$

只要输入序列长度 T 较小，隐藏维度 D 就主要决定自注意力和逐位置全连接层的复杂度。主要的限制因素是全连接层。但当输入序列变长时，序列长度 T 会逐渐主导这些模块的复杂度，使自注意力成为预训练语言模型的瓶颈。此外，自注意力的计算需要存储一个大小为 $T \times T$ 的注意力分数矩阵，这使得长输入序列的计算难以进行。因此，需要通过改进来降低长输入序列的计算开销。

为了将所有输入嵌入彼此连接起来，我们可以采用不同的模块。全连接层需要在不同嵌入之间建立 $T * T$ 个网络。卷积核宽度为 K 的卷积层无法连接所有成对的嵌入，因此在扩张卷积的情况下需要 $O(\log_K(T))$ 层。循环神经网络必须将网络应用 T 次。这便得出以下每层复杂度 [81, 141]

层类型	复杂度每层	顺序 最大值 运算	路径长度
自注意力	$O(T^2 * D)$	$O(1)$	$O(1)$
循环	$O(T * D^2)$	$O(T)$	$O(T)$
全连接	$O(T^2 * D^2)$	$O(1)$	$O(1)$
卷积层	$O(K * T * D^2)$	$O(1)$	$O(\log_K(T))$
受限自注意力	$O(R * T * D)$	$O(1)$	$O(T/R)$

最后一行描述了受限自注意力，其中自注意力仅考虑大小为 R 的邻域，以减少计算开销。显然，每层的计算复杂度是一个限制因素。此外，循环层的计算需要按顺序进行，无法并行化，如顺序操作一栏所示。最后一栏显示了路径长度，即在相距较远的位置之间传递信息所需的运算次数。输入序列与输出序列中任意位置组合之间的这些路径越短，就越容易学习长程依赖关系。与所有其他层类型相比，自注意力在此处具有明显优势。第 3.2 节讨论了用于处理更长输入序列的高级方法

总之，BERT 比其他层类型需要更少的计算开销。

2.1.7 总结

BERT 是一种自编码器模型，其主要任务是为词元生成上下文相关嵌入。在初步步骤中，训练数据中的单词和字母会被转换为词元，转换方式是让最常见的单词成为词元，并且任意单词都可以由词元组成。每个词元都通过输入嵌入进行编码。为标记每个输入词元的位置，会在输入嵌入中加入位置嵌入。

在 BERT 的每一层中，较低层的嵌入通过自注意力转换为新的嵌入。自注意力涉及计算嵌入的线性变换之间的标量积。这样，下一层的嵌入便能根据上下文中的词元进行调整，从而成为上下文相关嵌入。该运算会通过多个注意力头并行执行，各注意力头使用不同的线性投影。各注意力头可以针对不同的语义特征并行计算关联。随后，将得到的部分嵌入拼接成新的嵌入。除了自注意力头之外，每个编码器块还包含一个全连接层以及归一化操作。

原始 BERT 模型由六个编码器块组成，并为每个输入词元生成一个最终嵌入。BERT 在一个非常庞大的文档集合上进行预训练。主要的预训练任务是预测输入序列中被替换为 [MASK] 词元的单词。具体做法是将该词元最后一层的嵌入作为逻辑分类器的输入，由分类器预测该位置上各词元的概率。预训练过程中，模型参数通过随机梯度下降进行优化。这会促使模型在输出嵌入中收集有关该词元的所有可用信息。第一个输入词元是 [CLS] 词元。在预训练过程中，它用于预测下一句：以 [CLS]嵌入作为输入的逻辑分类器需要判断输入序列中的第一句和第二句是否属于同一段。

通常，预训练模型会使用一个规模较小的标注训练数据集，针对特定任务进行微调。一个示例是监督分类任务：判断输入文本表达的是正面、负面还是中性情感。同样，以 [CLS]嵌入作为输入的逻辑分类器需要确定这三种情感的概率。在预训练过程中，模型的所有参数都会进行微调。事实证明，这种迁移学习方法的准确率远高于仅使用小型训练数据集进行监督训练，因为模型可以利用在预训练过程中获得的语言知识。

实验表明，BERT 能显著提升许多语言理解任务的最先进水平，例如 GLUE 基准。其他应用包括命名实体识别，即在文本中识别人物、地点等名称；以及问答，即从一段

段落中提取问题的回答。对计算复杂度的分析表明，BERT 所需的计算开销低于其他层类型。总的来说，BERT 是自然语言处理领域的主力模型，并以不同变体用于解决语言理解问题。它的编码器块也被许多其他模型重复使用。

第3章介绍了提升 BERT 模型性能的方法，尤其是通过设计新的预训练任务（第3.1.1节）。第4章讨论了 BERT 模型所获得的知识。在第5至第7章中，我们介绍了 BERT 模型的一些应用，例如关系抽取（第5.4节）或文档检索（第6.1节）。

2.2 GPT：自回归语言模型

2.2.1 自回归语言模型的任务

为了捕获自然语言文本中的信息，可以通过语言模型描述词元的条件概率。这些自回归语言模型旨在根据文本中的前置词元预测下一个词元的概率。如果 V_{t+1} 是一个随机变量，其值为 v_{t+1} 在$t+1$位置处可能出现的词元，则我们需要计算条件概率分布 $p(V_{t+1}|v_1, \ldots\ldots, v_t)$。根据条件概率的定义，完整文本 $v_1, \ldots\ldots, v_T$ 的概率可以计算为

$$p(V_1=v_1,\ldots,V_T=v_T) = p(V_T=v_T|v_1,\ldots,v_{T-1}) * \cdots * p(V_1=v_1). \tag{2.9}$$

因此，条件概率可以表示有关有效句子的所有信息，包括语言使用得当与否。Qudar 等人 [115]提供了关于语言模型的最新综述。

在第 1.6 节中，我们使用循环神经网络构建语言模型。然而，这些模型难以确定词元之间的长程交互。作为替代方案，我们可以利用自注意力推断过去词元的上下文嵌入 $v_1, \ldots\ldots v_t$并基于这些嵌入预测下一个词元 v_{t+1}。

因此，我们需要将自注意力限制在词元 v_1、……、v_t上。这正是生成式预训练变换器 （*GPT*） [116, 118]所采用的方法。训练前，文本会转换为词元，例如采用字节对编码（第 1.2 节）。输入时，这些词元由词元嵌入和位置嵌入表示（第 2.1.1 节）。训练期间，GPT 模型以与 BERT 相同的方式执行第 2.1.1 节所述的自注意力计算。为了预测位置 $t+1$处不同词元的概率，自注意力仅限于先前的词元 v_1、……、v_t 及其嵌入。位置 $t+1$ 处可能出现的下一个词元的概率由逻辑分类器计算

$$p(V_{t+1}|v_1,\ldots,v_t) = \text{softmax}(A\tilde{\boldsymbol{x}}_{k,t} + \boldsymbol{b}), \tag{2.10}$$

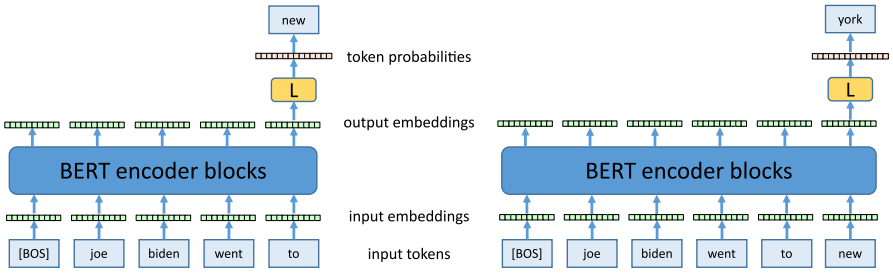


图 2.9 GPT模型的输入是词元 v_1、……、 v_t 直到位置 t的嵌入。GPT在不同层中计算这些词元的上下文自嵌入，并使用最高层中最后一个词元 v_t =“*to*” 的输出嵌入，通过逻辑分类器 L预测位置 $t+1$ 处可能词元的概率。对于实际观察到的词元“*new*”（左），该概率应当较高。随后，将观察到的词元 v_{t+1} =“*new*” 追加到输入序列中，并将其纳入自注意力计算，以预测位置 $t+2$处可能词元的概率；对于“*york*”（右），该概率应当较高。

该分类器以最后一层 k 在位置 t 处的嵌入 $\tilde{\boldsymbol{x}}_{k,t}$ 作为输入，用于预测位置 $t+1$ 处可能词元的随机变量 V_{t+1}（图 2.9）。这种方法称为**掩码自注意力**或**因果自注意力**，因为预测仅取决于先前的词元。由于 GPT 通过顺序应用同一个模型来生成词元，因此它被称为**自回归语言模型**。

2.2.2 通过预测下一个词元训练 GPT

训练目标适应了GPT的语言建模任务。图 2.9展示了两个连续词元的计算范围。通过教师强制，模型使用位置 t 之前观察到的词元 v_1、……、 v_t 计算自注意力，并预测下一个词元 v_{t+1}的词元概率。这由完整分布的因子分解（2.9）得到正当性说明。请注意，词元 v_s、 $s < t$的上下文嵌入会在每次将新词元 v_{t+1}、 v_{t+2}……纳入掩码自注意力计算时发生变化。由于GPT只考虑目标词元 v_{t+1}之前的词元，因此被称为**单向编码器**。 [3]给出了GPT的直观高层概览。

在训练过程中，必须通过优化来调整模型参数，使观测到的文档（2.9）的概率达到最大值。通过这种最大似然估计 （最大似然估计），可以针对大型文档语料库优化参数。为避免数值问题，可以通过最大化对数似然来求解，即对（2.9）取对数后求和

$$\log p(v_1,\ldots,v_T) = \log p(v_T|v_1,\ldots,v_{T-1}) + \cdots + \log p(v_2|v_1) + \log p(v_1). \quad (2.11)$$

或者，我们也可以最小化负对数似然 $-\log p(v_1, \ldots, v_T)$。

GPT-2 可以处理包含 1,024 个词元的输入序列，嵌入维度为 1024。在其 Medium 版本中，它拥有 3.45 亿个参数，包含 24 层，每层有 12 个注意力头。使用梯度下降进行训练时，批量大小为 512。该模型使用从 Reddit（一家社交媒体平台）抓取的 40GB 文本进行训练。仅纳入其他用户评价较高的文本，因此得到的数据集质量更高。较大的模型使用 256 个云端张量处理单元 v3 核心进行训练。训练时长以及训练的具体详细信息均未公开。

语言模型的质量可以通过给定文本集合 v_1、.....、 v_T 的概率 $p(v_1$、.....、$v_T)$ 来衡量。如果将其倒数按词元数量 T 进行归一化，就得到困惑度 [28]

$$ppl(v_1, \ldots, v_T) := p(v_1, \ldots, v_T)^{-\frac{1}{T}}. \tag{2.12}$$

较低的困惑度表示文本具有较高的概率。假设条件概率 $p(v_t|v_1$、.....、 $v_{t-1})$ 对所有 t 都相同，则得到 $ppl(v_1$、.....、 $v_T)$ $=1/p(v_t|v_1$、.....、 $v_{t-1})$，即下一个词元的逆概率。GPT-2 能够显著降低多个基准测试数据集上的困惑度，例如将宾州树库 [117] 上的困惑度从 46.5 降至 35.8，这意味着文本中的实际词语能够以更高的概率被预测出来。

可视化 GPT 嵌入

Kehlbeck 等人 [66] 研究了 BERT 和 GPT-2 在多变量空间中嵌入的相对位置，这两种模型各有 12 层。他们分别使用主成分分析（*PCA*） [111]和 UMAP [89]计算了三维投影。后者能够保留邻居的局部结构，但与 PCA 不同，它无法正确保持数据的全局结构。这些三维散点图可以在网站 [66]上进行交互式操作。结果表明，GPT-2 形成了两个彼此分离的簇：其中一个小簇只包含位于位置 0 的所有词元，而其他位置的嵌入则在第二个簇中形成带状结构。

仔细的研究表明，大多数嵌入向量都位于一个狭窄的锥体中，因此它们之间具有较高的余弦相似度 [25]。作者在上下文嵌入空间中识别出了孤立的簇和低维流形。Kehlbeck 等人 [66] 展示了，具有相同词性的词元在投影中会形成带状结构（图 2.8 左）。虚词都位于一个紧密的圆形结构上，而名词和动词等实词则位于其他延展结构中，并且与其他词性标签存在重叠。BERT 生成的嵌入形成了一个或多个簇（图 2.8 右）。虚词的嵌入彼此分离得相当明显，但名词、动词或形容词等实词则表现出一定的重叠。

位置> 0 处“banks” 和*“material”* 等实词的 GPT-2嵌入形成了拉长的带状结构，如图 2.10 右侧所示。对于更高

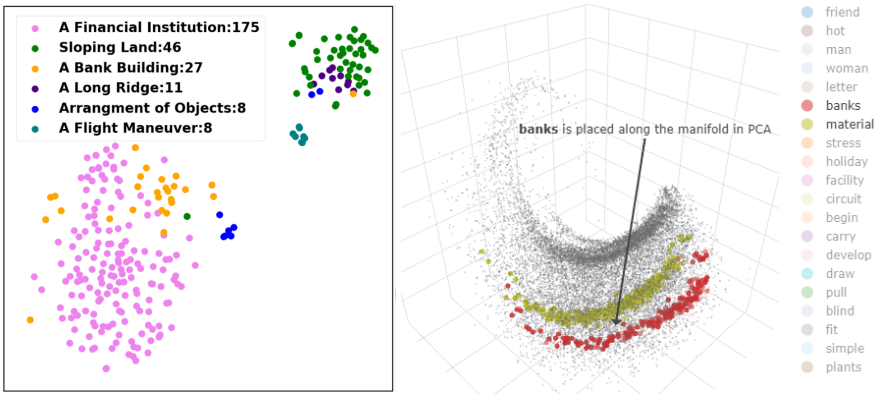


图 2.10 通过 t-SNE 将不同语义的 BERT 嵌入投影到二维空间后的图（左）。图例包含相应 WordNet 语义的简短描述以及其在训练数据中的出现频率 [153]。右侧展示了在不同上下文中计算的 *"banks"*（下方条带）和 *"material"*（中间条带）嵌入的 PCA 投影，以及其他词语的 PCA 投影。图像由交互方式生成，经作者许可后印刷，谨此致谢 [66]。

层，主成分分析投影会变得更加分散。用户可以通过指向每个点来读取词元上下文。

基于词元的自相似度 是指同一词元出现在不同句子中时的平均余弦相似度。在 BERT 和 GPT-2 中，实词的自相似度高于虚词 [66]。这可能表明，虚词在不同上下文中具有更多样的语义角色。评估某个词元关于余弦相似度的 10 个最近邻很有意义。在较低层中，对于这两个模型，最近的词元在大多数情况下都是相同的词元，只有少数实词例外。在较高层中，情况发生了变化，不同的词元成为最近邻。这表明，越高层的嵌入中包含的上下文信息越多。

作者还研究了由其他多种 PLM 类型生成的嵌入。他们发现，这些嵌入的结构截然不同，因为它们形成了不同的簇和流形。他们认为，在模型的新应用中，必须考虑这种结构。

2.2.3 生成词序列

训练后，GPT模型可以根据先前的词元 $t+1$ 预测下一个位置的词元概率 v_1、……、 v_t。要生成文本，我们必须根据这些概率选择一个词元序列。

- 随机采样根据预测概率选择下一个词元。这种方法有时会选中概率极低的词元，以至于整个句子的概率变得过低。尽管单个概率很小，但从低概率词元组成的群组中选取一个元素的概率却相当高。此外，对小概率的估计往往会受到错误的影响。

- *Top-k* 采样只考虑概率最高的 k 词元来生成下一个词元。概率质量会在这些词元之间重新分配 [42]，并用于随机选择一个词元。

- *Top-p* 采样会考虑累计概率超过阈值（例如 $p = 0.95$）的最小顶级候选项集合，然后根据重新分配后的概率选择下一个词元 [58]。这种方法会限制低频词元的概率质量，而这些词元会被忽略。

还有一些策略会通过降低更新公式 [67]中相应的得分，明确避免使用先前生成过的词元。Top-k 和 Top-p采样通常都能生成合理的词元序列，实际上也被用于生成文本。

有多种方法可以改进词元选择。Meister 等人 [90]发现，人类生成的文本往往具有均匀分布的“惊奇度”。这意味着，平均而言，下一个词元既不应过于罕见，也不应过于常见。他们提出了多种采样准则，例如方差正则项。

Martins 等人 [86] 认为，Softmax 生成的输出分布并不现实，因为它为每个输出词元都分配了正概率。他们提出了*Entmax* 变换，该变换根据计算得到的得分生成稀疏概率分布，其中部分概率恰好为零。Entmax 变换可以通过参数 $\alpha \geq 1$进行控制。对于 $\alpha = 1$ ，我们得到 Softmax；而 $\alpha = \infty$ 则恢复为 argmax。对于中间值 $\infty > \alpha > 1$。0，某些词元会获得恰好为零的概率。Entmax 损失是凸且可微的，因此可以通过反向传播进行训练。与 top-p 采样一样，并且不同于 top-k采样，Entmax 采样会根据上下文考虑数量不固定的词元。实验表明，与其他方法相比，Entmax 能带来更好的困惑度和更少的重复。与 top-p 采样相比，它所考虑的词元数量变化更大。

Khandelwal 等人 [68] 尝试利用词元 n-gram 的统计信息来改进语言模型的估计概率。他们会对已经处理的最后若干词元执行最近邻搜索。他们使用预训练嵌入空间中的距离作为距离度量。从检索到的最近邻中，他们获得关于可能的下一个词元的额外证据，并将其与语言模型的词元概率合并。通过这种方式，他们能够改善语言模型的困惑度。该方法对于预测稀有模式尤其有帮助，例如事实知识。

Yang等人 [157] 分析了 softmax 函数的性质。他们发现，标准softmax 的容量不足以建模自然语言，因为它限制了从映射到概率的秩。他们提出通过

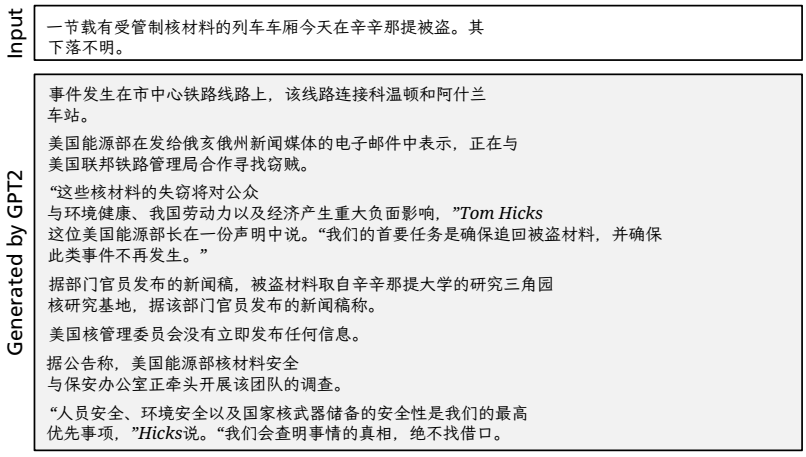


图 2.11 给定输入文本后，GPT-2 通过 top-k 采样 [117]生成续写。经作者许可引用。

*Softmax*混合来预测概率，即对多个不同逻辑分类器进行凸组合；这种方法比单个 softmax 的表达能力更强。作者表明，这一改动改善了语言建模中的困惑度，也提升了其他 Transformer 架构的性能 [101]。

2.2.4 高级语言模型 GPT-2

GPT-2 [118]是首个能够生成语法正确、语义合理文本文档的语言模型。其最大版本拥有 48 个编码器块，包含 15 亿个参数，可处理长度为 1,600 个词元的序列。给定一段初始文本，模型会适应该文本的风格和内容，并生成回答；生成的续写往往难以与人工创作的续写区分开来。不过，较长的生成文本有时会显得重复，连贯性也较差。

GPT-2 使用 top-k 截断采样来生成图 2.11 所示的示例文本 [117]。可以看出，文本没有语法错误，生成的内容也合乎情理。作者指出，二分之一的试验生成了高质量文本。模型会适应输入文本的风格和内容，因此用户可以生成围绕自己感兴趣主题的逼真且连贯的续写。显然，该主题必须在 Reddit 训练数据中有所提及；这些数据涵盖了广泛的主题，例如新闻、音乐、游戏、体育、科学、烹饪和宠物。

该模型无需针对特定任务进行训练，就能比以往的模型更好地完成许多任务。这种学习方式称为零样本学习。例如，GPT-2 在宾夕法尼亚树库测试集上的困惑度为 35.8

相比之下，此前较差的最先进水平为 46.5 [117]。这一成果是在未使用宾州树库语料库 [135]训练 GPT-2 的情况下取得的。

2.2.5 微调 GPT

通过微调，GPT-2 可以适应新类型的文本，例如新的文本体裁。例如，为了创作歌词，St-Amant [4] 使用包含 12,500 首英语摇滚歌曲歌词的数据集，对 GPT-2 进行 5 个训练轮次的微调。随后，模型便能够续写流行歌曲的歌词，而这些歌词在训练期间模型从未见过。该模型应用于歌词时，BLEU分数高达 68。另一项实验介绍了诗歌的生成 [19]。

与 BERT 类似，预训练 GPT-2 也可以经过修改来执行分类任务。一个示例是微调模型，对文档的情感进行分类，判断其为正面还是负面。Radford 等人 [116] 将分类任务编码为带有特定词元和最终结束标记 [结束]的文本。随后，模型需要预测该序列。最高层中 [结束] 的嵌入被用作逻辑分类器的输入，该分类器经过训练，用于预测各类别的概率。作者发现，将微调数据的语言建模（2.11）作为微调的辅助目标，有助于提升泛化能力并加快收敛。他们将 GLUE（第 2.1.5 节）得分从 68.9 提高到 72.8，并在 8 项自然语言理解 GLUE 任务中的 7 项达到最先进水平。结果表明，语言模型能够捕捉与句法和语义相关的信息。

然而，GPT 在预测下一个词元时是从左到右运行的。在句子 *“I went to thebank to deposit cash”* 和 *“I went to the bank to sit down”* 中，它在预测 *“sit”* 或 *“deposit”* 时，会为 *“bank”* 创建相同的上下文相关嵌入，尽管词元 *“bank”* 在这两个上下文中的意义不同。相比之下，BERT 具有双向性，在预测掩码词元时会考虑文本中的所有词元。这一事实解释了为什么 BERT 在某些任务上表现出更好的性能。

2.2.6 总结

GPT采用了类似BERT模型的架构，逐个生成句子中的词元。它从一个词元输入序列开始，该序列也可以为空。词元编码为词元嵌入与位置嵌入之和。GPT使用与BERT相同的编码器块，但计算过程采用掩码，即仅限于已生成的词元。对于这些词元，模型会在多个层中生成上下文嵌入。顶层最后一个词元的嵌入会输入逻辑分类器，由它计算下一个位置上各词元的概率。随后，将观察到的词元追加到

下一个位置的输入中，并针对再下一个位置重复计算。因此，GPT被称为自回归语言模型。

训练过程中，随机梯度下降会调整参数，使模型能够为训练数据中观察到的词元预测较高的概率。这里采用最大似然准则，以优化输入数据的概率。模型在大型文本数据集上完成训练后，便可投入使用。给定起始文本，模型可以依次计算下一个词元的概率，然后根据这些概率选出一个新词元。

如果考虑所有备选词元，通常会选出稀有词元。通常会将符合条件的词元数量限制为 k 个高概率词元（top-k 采样），或者只纳入高概率词元，直至其概率总和达到指定值 p （top-p 采样）。这样生成的文本会好得多。GPT-2 等高级语言模型拥有数十亿个参数，能够生成合乎情理且没有语法错误的故事。

GPT模型也可以进行微调。第一种微调方式是让模型适应特定的文本体裁，例如诗歌。另一种方式是将 GPT 用作分类器：将针对输入文本最近生成的词元的输出嵌入输入逻辑分类器。采用这种方法，GPT-2 在 GLUE 基准测试中的大多数自然语言理解任务上都达到了更先进的水平。这表明 GPT-2 已经掌握了关于语言的全面知识。然而，由于自注意力只能关注过去的词元，因此 BERT 等模型可能表现更好，因为它们在计算过程中可以考虑所有输入词元。

第3章讨论了如何提升 GPT模型的性能，尤其是通过使用更多参数（第3.1.2节）。这些拥有数十亿参数的大型模型，无需微调便可按指令执行多种任务（第3.6.3节）。在第5至第7章中，我们介绍了 GPT 模型的多种应用，例如问答（第6.2.3节）、故事生成（第6.5节），以及根据文本生成图像（第7.2.6节）。

2.3 Transformer：序列到序列翻译

2.3.1 Transformer 架构

基于循环神经网络（第1.6节）的翻译模型受到循环神经网络顺序特性的限制。要确定词元 v_s 与 v_t 之间关系所需的运算次数，会随着位置之间的距离 $t - s$ 增加。模型必须同时将所有词元之间的关系存储在一个向量中，因此难以学习相距较远位置之间的复杂依赖关系。

与循环神经网络翻译模型类似，*Transformer* [141] –基于编码器和解码器模块（图 2.13）。编码器与 BERT 非常相似，而解码器则类似于 GPT。它是一种序列到序列模型（序列到序列），可将输入语言的源文本翻译为目标语言中的目标

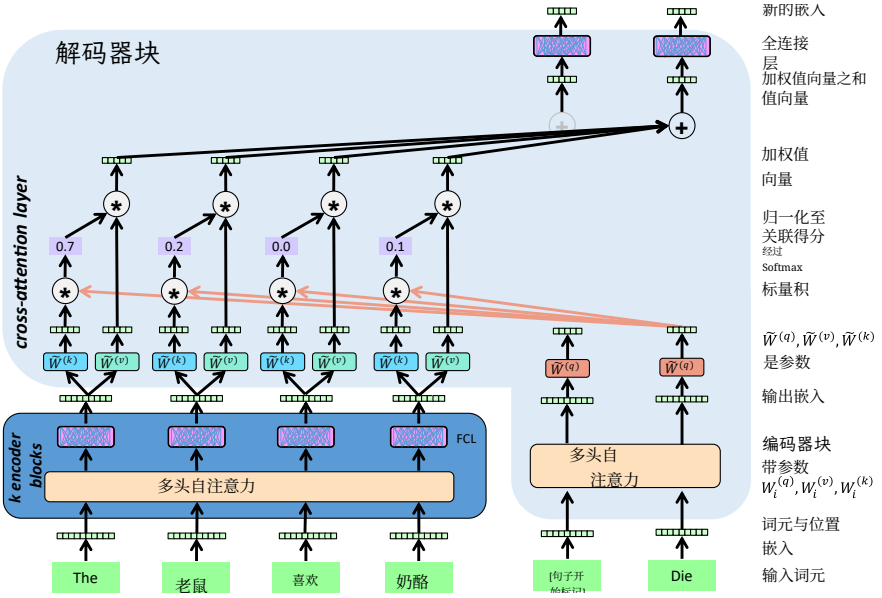


图 2.12 Transformer [141] 使用 k 与 BERT（图 2.4）具有相同架构的编码器块，为输入文本的所有词元生成上下文嵌入。解码器块是一个自回归语言模型（图 2.9），按顺序预测目标语言中的下一个词元。每个编码器块都包含一个针对当前输出词元序列的多头自注意力机制。通过交叉注意力，可以纳入输入序列中的信息。针对所有当前输入词元重复这些计算，其过程与自注意力计算非常相似。所得向量经过全连接层转换，生成该层的嵌入。

语言。它不需要通过大量计算步骤来关联相距较远的词元，而是直接在一个步骤中并行计算这些词元之间的自注意力。

编码器 生成源文本词元 $\tilde{\boldsymbol{x}}_1$、……、$\tilde{\boldsymbol{x}}_{T_{\text{src}}}$ 的上下文嵌入，其架构与BERT模型（图 2.4）完全相同。原始 Transformer [141] 使用 6 个编码器块。最后一层生成的嵌入表示为 $\check{\boldsymbol{x}}_1$、……、$\check{\boldsymbol{x}}_{T_{\text{src}}}$。

Transformer 解码器逐步计算目标词元的概率分布 $p(S_t | s_1, \ldots, s_{t-1}, v_1, \ldots, v_{T_{\text{src}}})$，与循环神经网络类似。请注意，源词元 v_i 以及已观察到的目标词元 s_j 都作为条件。根据条件概率的定义，这便得到了输出分布的总概率

$$\begin{aligned} & p(S_1 = s_1, \ldots, S_T = s_T | v_1, \ldots, v_{T_{\text{src}}}) \\ & = p(S_T = s_T | s_1, \ldots, s_{T-1}, v_1, \ldots, v_{T_{\text{src}}}) \cdots p(S_1 = s_1 | v_1, \ldots, v_{T_{\text{src}}}), \end{aligned} \tag{2.13}$$

其中，S_t 是一个随机变量，其值为位置 t 处可能的目标词元 s_t。训练过程中会最大化该概率。

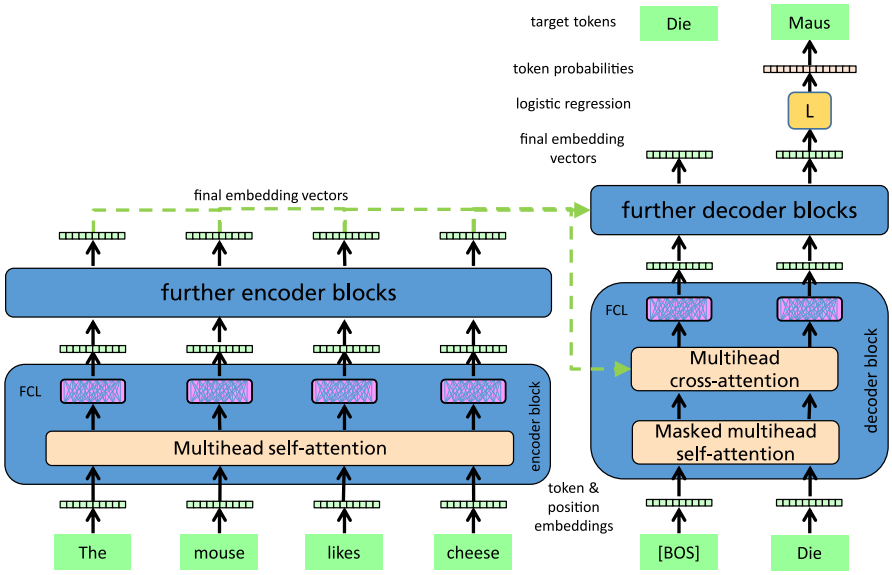


图 2.13 Transformer [141] 使用与 BERT 架构相同的编码器，为输入句子中的所有词元生成嵌入。每个编码器块都会对输入序列执行多头自注意力运算，然后通过一个全连接层（FCL）。解码器类似于 GPT模型，并按顺序预测目标语言中的下一个词元。每个解码器块都包含多头交叉注意力，其中纳入了编码器的最终嵌入。逻辑分类器 L 使用最终解码器块的最后一个输出嵌入，预测输出句子中下一个词元的概率。

我们用 s_0、s_1、……、s_{t-1} 表示已经翻译的词元，其中 s_0 是词元“[句子开始标记]”，表示输出文本开头。解码器首先使用公式（2.4）对这些词元计算自注意力，该过程与 BERT 中的过程相同。由于只有部分目标词元被覆盖，其余部分为“掩码”，因此这一层称为**掩码多头自注意力**，并为目标词元 s_0、s_1、……、s_{t-1}生成中间的上下文嵌入 $\tilde{\boldsymbol{s}}_0$、$\tilde{\boldsymbol{s}}_1$、……、$\tilde{\boldsymbol{s}}_{t-1}$。

交叉注意力

随后，解码器使用最高层编码器块的输入文本嵌入执行交叉注意力 Catl($\tilde{\boldsymbol{V}}$，$\check{\boldsymbol{X}}$)（图 2.12）。在此过程中，针对相应解码器块提供的目标词元 $\boldsymbol{S}_t = (\tilde{\boldsymbol{s}}_0, \tilde{\boldsymbol{s}}_1, \ldots, \tilde{\boldsymbol{s}}_{t-1})$ 的嵌入计算查询向量 $\tilde{}$。针对最后一个编码器块的嵌入 $\check{\boldsymbol{X}} = \check{\boldsymbol{x}}_1, \ldots, \check{\boldsymbol{x}}_{T_{\text{src}}}$ 计算键向量和值向量。请注意，交叉注意力使用与 BERT 自注意力相同的公式 (2.4)，并采用矩阵 $\boldsymbol{W}_{(q)}$，$\boldsymbol{W}_{(k)}$，$\boldsymbol{W}_{(v)}$。这一过程并行执行，称为多头交叉注意力。通过这种方式，信息

会将源文本纳入考量。随后，将不同注意力头计算出的嵌入拼接起来 (2.6)，再通过带有 ReLU 激活的全连接层 (2.7) 对结果进行变换。此外，还使用残差“旁路”连接和层归一化 [6] 进行正则化。全连接层的输出会为目标词元$s_1, \ldots, s_{t-1}$生成新的“输出”嵌入 $\tilde{\boldsymbol{s}}_0, \ldots, \tilde{\boldsymbol{s}}_{t-1}$。这些层合称为解码器块（图 2.13）。

下一个解码器块将上一个块计算得到的词元输出嵌入作为输入，并计算目标词元 s_1、……、s_{t-1} 的新嵌入。解码器由多个解码器块组成（原始模型中有 6 个）。使用最后一个解码器块中最右侧词元 s_{t-1} 的输出嵌入 $\tilde{\boldsymbol{s}}_{t-1}$，逻辑分类器会预测目标文本在位置 t 处的下一个词元 s_t 的词元概率 $p(S_t = s_t | s_1、\ldots\ldots、s_{t-1}, v_1、\ldots\ldots、v_{T_{\text{src}}})$，例如图 2.13 中的词元 *“Maus”*。

请注意，要预测位置 $t+1$ 处的后续词元时，会将观测到的词元 s_t 加入解码器自注意力的计算（2.13）中。因此，解码器嵌入会发生变化，所有解码器计算都必须重新执行。在这方面，模型仍以递归方式运行。尽管如此，每一层中的所有自注意力和交叉注意力都是并行计算的。不过，编码器的计算只执行一次。

不同长度的序列会用特殊词元填充至最大长度。输入序列和输出序列都要这样处理。如果序列很短，就会浪费大量空间。因此，在训练数据中，可以让不同小批次（称为桶）采用不同的序列长度。

Transformer包含大量参数。首先，需要为输入词元和目标词元的词汇表生成嵌入。接下来，还需要为不同注意力头和层中的多头自注意力、掩码多头自注意力以及多头交叉注意力设置 $\boldsymbol{W}_{(q)}$、$\boldsymbol{W}_{(k)}$、$\boldsymbol{W}_{(v)}$ 矩阵。此外，还必须指定全连接网络和最终逻辑分类器的参数。基础模型的输入序列长度为512，含有6500万个参数；大型模型的输入序列长度为1,024，含有2.13亿个参数 [141]。所有这些参数的值都会在训练过程中进行优化。

训练数据由输入句子与对应目标句子组成。训练的目标是针对给定的输入词元，通过随机梯度下降，以最大概率生成目标词元，从而最大化输出序列的联合条件概率（2.13）。在图2.13的示例中，对于给定的输入文本*“The mouse likes cheese”*，必须最大化输出词元*“Die Mausmag Käse”* 的条件概率乘积。例如，原始模型[141], 使用了WMT英法语基准数据中的3600万个句子，并将其编码为32,000个WordPiece词元。编码器和解码器通过随机梯度下降同时进行端到端训练，使用8个图形处理器需耗时3.5天。

交叉注意力是 Transformer 的核心部分，它将输入句子中的信息与翻译后的输出句子联系起来。图2.14展示了一个德语输入句子及其英语翻译。两个句子都经过字节对编码分词，其中单词的开头用“_”表示。

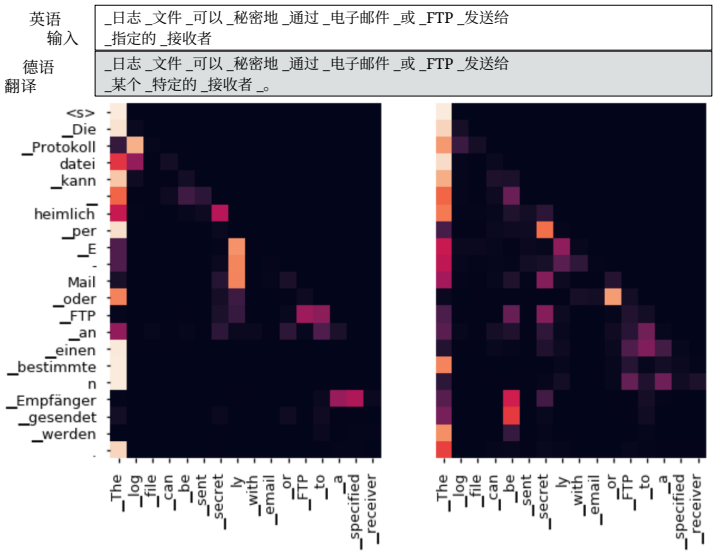


图2.14 经过字节对编码分词的英语输入句子，以及翻译后经过分词的德语输出句子。下方是来自第 4 个解码器层的不同注意力头的两张交叉注意力图 [126]。较暗的值表示较低的交叉注意力分数。图像来源：[126]。

下图展示了两个不同注意力头中输入词元与输出词元之间交叉注意力的强度。显然，第一个输入词元“*_The*”具有特殊作用。

2.3.2 解码翻译以生成词语

训练后，Transformer 能够预测输入句子的输出词元概率。然而，在实际翻译中，必须生成明确的输出词元序列。计算概率最大的输出序列在计算上十分困难，因为这需要考虑所有可能的输出序列。因此，可以使用贪心解码或束搜索获得近似解。

贪心解码会在每个解码步骤中，直接选取概率最高、最有可能的词元，直到生成句末标记。这种方法的问题在于，每当某个时间步 t选定输出后，就无法再返回并更改选择。实际上，贪心解码往往会遇到问题，因为可用的高概率后续词元可能与先前分配的词元不匹配。由于无法修改决策，这可能导致生成的译文并非最优。

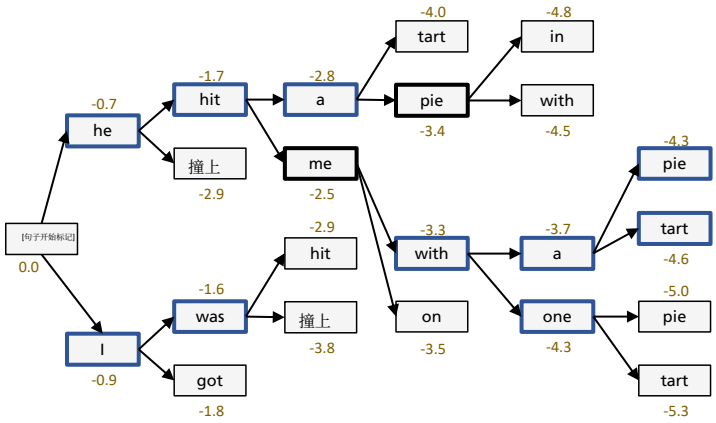


图 2.15 束搜索是一种用于语言模型解码并生成文本的技术。在每一步中，算法都会跟踪 k 概率最高的部分翻译（粗体间隔）。每个翻译的得分等于其对数概率。束搜索会持续进行，直到每个分支都到达结束标记 [78]。

束搜索 [52] 会保留固定数量 k 的可能译文 s_1、……、 s_t，这些译文的长度会逐渐增加（图 2.15）。每一步中，长度为 t 的每个译文都会在位置 $t+1$ 扩展出 k个不同的词元，这些词元具有最高的条件概率 p（S_{t+1} =$s_{t+1}|s_1$、……、s_t， v_1、……、$v_{T_{\text{src}}}$）。在这些 $k*k$ 个词元序列中，只保留总概率 p最高的 k 个序列（s_1、……、 $s_{t+1}|v_1$、……、$v_{T_{\text{src}}}$）。完整译文（包含句末标记）会被加入最终候选列表。随后，算法从该列表中选取概率最高的译文（按目标词数量归一化）。当 $k=1$ 时，束搜索会退化为贪心解码。实际上，束搜索（束宽为 4）所得的翻译质量显著优于贪心解码。较大的束宽往往会导致次优解 [31]。不过，与贪心解码相比，束搜索的计算成本非常高（速度慢 25% 至 50%，具体取决于基础架构和束宽） [29]。

2.3.3 翻译评估

传统上，评估是通过将一个或多个参考译文与生成译文进行比较来完成的，正如综述 [127] 中所述。自动评估指标有很多：

BLEU 比较词元的 1 元语法至 4 元语法计数。BLEU 指标的取值范围为 0 到 1，其中 1 表示输出与参考译文完全相同。尽管 BLEU 与人工判断相关性良好 [110], ，但它仅依赖精确率，不考虑召回率——即参考译文中匹配的 n 元语法数量占参考译文中 n 元语法总数的比例。

ROUGE [80] 与 BLEU 不同，ROUGE 是一种基于召回率的度量，用于确定参考文本中的单词或 n 元语法有多少比例出现在生成文本中。它还可以衡量其他内容，例如一对文本之间的一元语法或二元语法的重叠情况，以及最长公共子序列。ROUGE 有多个版本：ROUGE-1 衡量一对文本之间一元语法（单个单词）的重叠情况。ROUGE-2 衡量一对文本之间二元语法（由两个词语构成的序列）的重叠情况。ROUGE-L 衡量两段文本共有的最长词语序列的长度（不一定连续，但顺序不变）。该长度除以参考文本中的词语数量。

METEOR [75] 旨在解决 BLEU 的不足。它在翻译输出与给定参考译文之间执行逐词对齐。对齐过程通过一系列词映射模块生成。这些模块会检查词语是否完全相同、使用 Porter 词干提取器进行词干还原后是否相同，以及彼此是否为同义词。在获得最终对齐结果后，METEOR 会计算 F 值，即单词精确率和召回率的参数化调和平均数。METEOR 还被证明与人工判断具有较高的相关性，通常甚至优于 BLEU。

BERTScore [164] 会考虑同义词，并衡量翻译文本与参考译文之间嵌入的相似度。它计算两段文本中所有词元嵌入之间的余弦相似度。随后使用贪心匹配方法来确定词元的分配。最大分配相似度即为 BERTScore。

然而，对于高质量翻译而言，人工判断与自动评估之间存在明显差异。因此，如今大多数高端比较都会采用人类专家来评估翻译质量以及其他文本生成方法。自 Vaswani 等人在 2017年提出 Transformer 以来，其变体得以提升语言翻译性能的最先进水平，例如在 WMT2014 英法语翻译任务中，BLEU分数从 37.5 提升至 46.4。

Transformer 架构得到了理论分析。Yun 等人 [160] [161]证明，Transformer 的表达能力足以捕捉定义域紧致的所有连续序列到序列函数。Pérez 等人 [112] 推导出，完整的 Transformer 是图灵完备的，也就是说，它可以模拟完整的图灵机。

2.3.4 预训练语言模型与基础模型

模型语言模型 要么计算自然语言文本的联合概率，要么计算条件概率，并且可能包含有关该语言的所有信息。BERT 是一种自编码器 语言模型，包含用于生成词元上下文嵌入的编码器块。GPT 是一种自回归语言模型，用于预测序列的下一个词元，并将自注意力限制在已经生成的词元上。*Transformer*（或 *Transformer* 编码器-解码器）使用 Transformer 编码器将输入文本转换为上下文嵌入，并利用以编码器嵌入为输入的自回归 Transformer 解码器生成译文

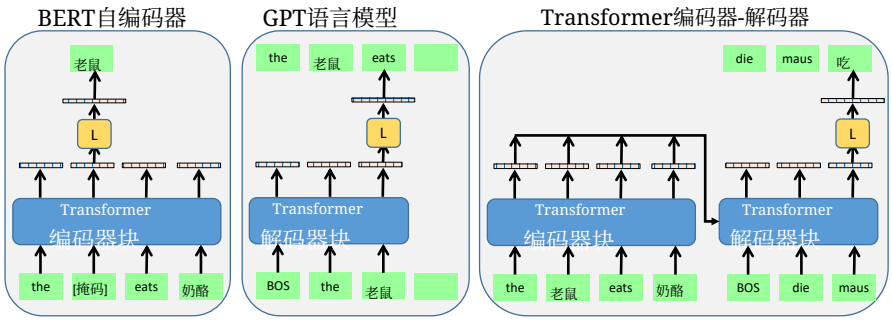


图 2.16 像 BERT（左）这样的自编码器和像 GPT-2（中）这样的自回归语言模型使用 Transformer 模块来生成词元的上下文嵌入。Transformer（右）结合了 Transformer 编码器和自回归 Transformer 解码器来生成翻译。所有模型都使用逻辑分类器预测词元的概率 L。这些模型统称为预训练语言模型（PLM）。

（图 2.16）。这些模型是现代自然语言处理的骨干网络，统称为预训练语言模型（*PLM*）。

所有这些模型，尤其是 BERT 和 GPT，都是通过在大型文本文件语料库上进行预训练来初始化的。在预训练期间，输入中的部分内容会被隐藏，模型经过训练以重建这些部分。这种方法已被证明在构建强大的语言表示，以及为可适配特定任务的高表达能力自然语言处理模型寻找参数初始化方面极为有效。最后，这些模型提供了语言的概率分布，我们可以从中采样。

大多数网络类型都带有一些内置假设，称为归纳偏置。卷积网络具有局部核函数，这些函数在输入矩阵上移动，因此具有平移不变性和局部性的归纳偏置。循环网络将同一网络应用于每个输入位置，因此具有时间不变性和局部性。BERT 架构对数据中的结构依赖只作了少量假设。GPT模型与循环神经网络相似，因为它假设依赖关系具有马尔可夫结构，并据此预测下一个词元。因此，预训练语言模型通常需要更多训练数据来学习不同数据点之间的交互，但之后能够比其他类型的模型更准确地表示这些交互。

从历史上看，学习得到的嵌入向量被用作下游任务中的词表示（图 2.17）。早在 2003年，Bengio 等人 [15]就提出了词的分布式向量表示，通过循环模型预测下一个词。2011年，Collobert 等人 [32]成功将词嵌入应用于词性标注、组块分析、命名实体识别和语义角色标注。2013年，Mikolov 等人 [93] 使用逻辑分类器推导出词嵌入。2015年，Dai 等人 [33] 通过自监督方式，使用 RNN语言模型训练嵌入，随后将其应用于文本分类。2017年，McCann 等人

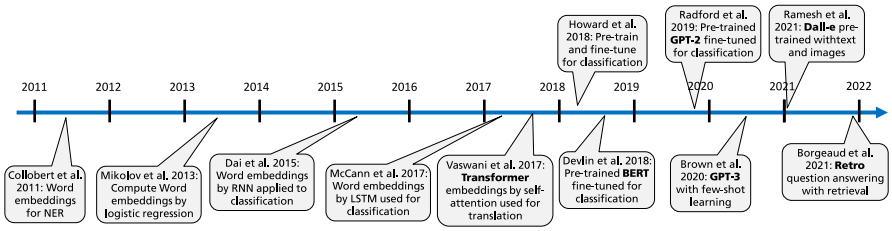


图 2.17 嵌入、预训练和微调的发展时间线。

al. [87] 使用预训练的多层长短期记忆网络计算上下文化词向量，随后将其用于各种分类任务。

同年，Vaswani 等人 [141]开发了用于语言翻译的纯注意力 Transformer。2018年，Howard 等人 [59] 预训练了一个语言模型（ULMFiT），并通过针对每项任务更新完整的（预训练）模型，展示了微调 [116] 不同目标任务的有效性。同年，Radford 等人 [141] 使用 Transformer 的预训练自回归部分 [39]，通过微调模型解决了大量文本理解问题。与此同时，Devlin 等人 [118]使用掩码语言模型目标预训练了自编码器，并通过微调将这一 BERT模型应用于许多下游任务。2019年，Radford 等人 [21] 推出了 GPT-2 语言模型，该模型能够生成语义上令人信服的文本。Brown 等人 [121] 提出了 GPT-3 模型，该模型可以根据任务描述和一些示例来执行自然语言处理任务。2021年，Ramesh 等人 [18] 将语言建模应用于文本和图片，能够根据文本描述创作令人印象深刻的图片。Borgeaud 等人提出了 Retro 模型，该模型通过从包含 2 万亿个词元的文本集合中检索信息来回答问题，并以自然语言组织回答。

如今，几乎所有最先进的自然语言处理模型都由少数几种预训练语言模型改造而来，例如 BERT、GPT-2、T5 等。预训练语言模型正变得越来越庞大、越来越强大，带来了新的突破，也吸引了越来越多研究团队的关注。由于性能大幅提升，一些研究团队建议将大规模预训练语言模型称为基础模型，因为它们构成了一项具有「基础性」意义的突破性技术，可能会影响多种类型的应用 [17, 页。 3]。在本书中，我们将「基础模型」一词专用于参数超过十亿的大型预训练语言模型，因为这些模型能够生成流畅文本，可能处理不同类型的媒体，而且通常可以通过提示来执行特定任务。

如果其中一个模型得到改进，这种高度同质化就能为许多自然语言处理应用带来立竿见影的收益。另一方面，所有系统也可能共同存在少数几个基础模型中的问题偏见。正如我们将在后续章节中看到的，基于预训练语言模型的序列建模方法现已应用于文本（第 2.2 节）、语音（第 7.1 节）、图像（第 7.2 节）、视频（第 7.3 节）、计算机代码

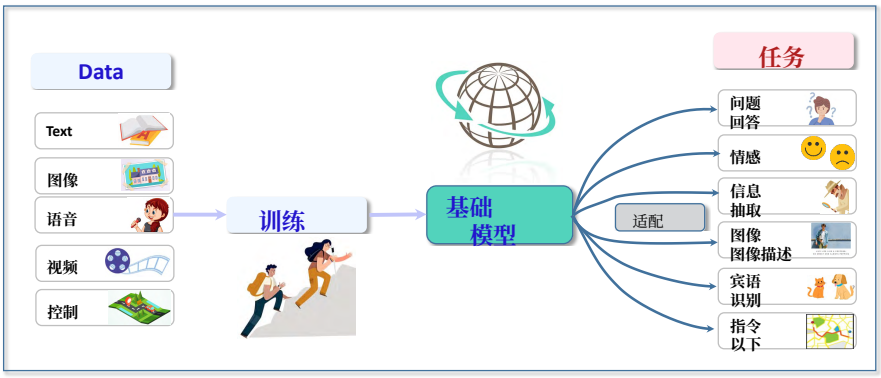


图 2.18 基础模型可以整合来自不同模态的数据中的信息。随后，可以通过微调等方式，将其适配到广泛的下游任务中 [17, 第 6]页。表 A.1 中图像部分的版权信息。

（第 6.5.6 节）以及控制（第 7.4 节）。图 2.18 展示了基础模型的这些广泛能力。

下一节 2.4 将讨论优化和正则化预训练语言模型的一些常用技术。此外，还将介绍一些用于修改这些网络架构的方法。在第3章中，我们将介绍若干提升预训练语言模型能力的方法，尤其是通过修改训练任务来实现的方法（第 3.1.3 节）。在第5至第7章中，我们将讨论预训练语言模型的若干应用。第5章介绍传统的自然语言处理任务，例如命名实体识别和关系抽取，目前预训练语言模型在这些任务上的表现最佳。基础模型最重要的应用，一方面是文本生成及相关任务，例如问答和对话系统，这些内容将在第6章介绍。另一方面，基础模型可以同时处理不同的媒体，并执行图像描述、图像中的目标检测、根据文本描述进行图像生成、视频理解或电子游戏控制等任务，这些内容将在第7章讨论。鉴于此类基础模型可能带来的社会和社会群体层面的影响，研究人员在开发和应用这些模型时，尤其需要牢记社会价值观和人权。这些方面将在第 8.2 节中进行总结。

可用实现

- 许多预训练语言模型（BERT、GPT、Transformer）的源代码，以及适用于不同语言和文本语料库的预训练模型，均可从 Hugging Face https://huggingface.co/transformers/、Fairseq https://github.com/pytorch/fairseq、TensorFlow https://www.tensorflow.org/ 和 PyTorch https://pytorch.org/ 下载。这些工具包还支持灵活构建深度神经网络，并

提供梯度的自动计算以及优化方法。所有这些工具包都能并行执行计算，并将计算分配到不同的 CPU 和图形处理单元（GPU）上。

- 预训练语言模型的规模越来越大，超出了单个 GPU 的内存容量，因此需要在多个 GPU 之间分配训练代码。FastSeq https://github.com/microsoft/fastseq、LightSeq https://github.com/bytedance/lightseq 和 FastT5 https://github.com/Ki6an/fastT5 等库支持这一功能。
- DeepSpeed [122] 被用于训练具有 530B 参数的 MT-NLG 自回归语言模型（第 3.1.2 节）https://github.com/microsoft/DeepSpeed。
- Ecco [2] https://github.com/jalammar/ecco 和 BertViz [144]https://github.com/jessevig/bertviz 是用于可视化预训练语言模型注意力和嵌入的工具。
- Transformers-interpretlittps://github.com/cdpierse/transformers-interpret 是为 Hugging Face 软件包设计的模型解释工具。
- Captum [70]是一个用于生成 PyTorch 模型预测结果的解释结果和解释的库 https://captum.ai/ 。

2.3.5 总结

Transformer 是一种序列到序列模型，可将输入语言的源文本翻译为目标语言的目标文本。它由一个编码器组成，其架构与自编码器 BERT 模型相同，用于计算源文本中词元的上下文嵌入。解码器类似于自回归 GPT 模型，并按顺序生成目标文本的词元。在内部，不同层会计算目标词元的上下文嵌入。每个解码器块都额外包含一个交叉注意力模块，其中查询向量取自目标词元的嵌入，而键向量和值向量则根据最后一层中源词元的嵌入计算得到。通过这种方式，源文本中的信息被传递给解码器。顶层中最后一个词元的嵌入会输入逻辑分类器，由该分类器计算下一个位置各词元的概率。随后，将下一个位置观测到的词元追加到目标输入中，并重复这些计算，以确定再下一个位置的词元。

在训练过程中，通过随机梯度下降调整 Transformer 的参数，使模型为训练数据中观察到的翻译目标词元分配较高的概率。当模型在大型文本数据集上完成训练后，便可用于翻译。给定输入文本，模型可以顺序计算翻译中下一个词元的概率。

应用训练好的模型时，要么选取概率最高的词元，要么通过束搜索生成多个候选项，并选择概率最高的最终输出序列。由于正确的译文可能不止一种，因此翻译质量的评估颇为困难。人们开发了多种指标，例如 BLEU，通过比较机器翻译与一条或多条参考译文中共同的 n-元词组数量来进行评估

$n = 1, \ldots\ldots, 4$。结果通常由人工评估者进行评估。Transformer 能够生成比先前模型更好的译文。如今，对于多种语言对而言，机器翻译的质量已达到与人工译者相当的水平。

在前面的章节中，我们讨论了自编码器 *BERT* 模型、自回归 *GPT* 模型以及编码器-解码器 *Transformer*。这些模型统称为预训练语言模型，因为采用迁移学习，即先使用大型训练集进行预训练，随后进行微调，是这三类模型的核心方法。自注意力和交叉注意力模块是所有三类模型所使用的核心构件。尽管近年来发展出了许多变体，Vaswani 等人提出的原始架构 [141] 仍被广泛采用。

事实证明，这些模型不仅可以应用于文本，还可以应用于各种类型的序列，例如图像、语音和视频。此外，通过简单的提示，还可以指示它们执行各种任务。因此，大型预训练语言模型也被称为基础模型，因为人们期望它们在未来文本和多媒体系统的发展中发挥关键作用。

2.4 预训练语言模型的训练与评估

本节介绍训练和应用预训练语言模型所需的一些技术。

- 我们需要能够处理数百万乃至数十亿参数和训练示例的优化技术。
- 训练模型并避免过拟合需要特定的正则化方法。
- 必须估计模型预测的不确定性，以评估模型的性能。
- 解释模型预测对于提高模型的接受度非常有帮助。

本节将讨论解决这些问题的方法。预训练语言模型通常使用当前的某种深度学习框架来实现。最受欢迎的框架是 Google 提供的*Ten-sorFlow* [137] 和 Meta 提供的*PyTorch* [114]。两者都基于 Python编程语言，并包含用于定义网络、在专用硬件上并行训练网络，以及将其部署到不同环境中的语言元素。新兴的*JAX* 框架 [22], 在快速实验方面尤其灵活。它是一种线性代数编译器，可加速机器学习研究中的计算。

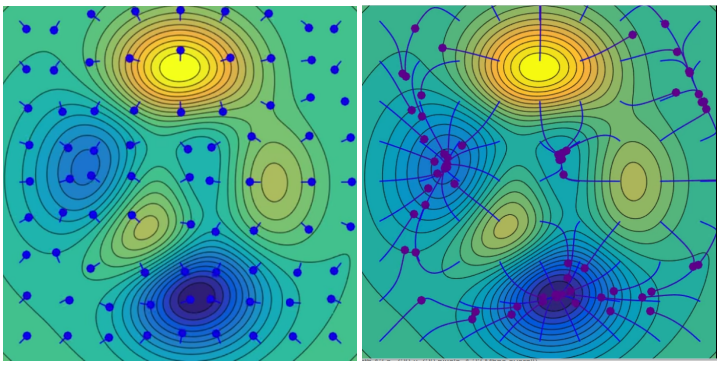


图 2.19 在该二维函数 $L(\boldsymbol{w})$（左图）的网格各点上计算负梯度。梯度下降算法沿着负梯度方向移动，并逐渐接近局部极小值（右图）。蓝线表示最小化过程中采取的路径。图像来源见表 A.1。

2.4.1 预训练语言模型优化

预训练语言模型优化基础

对于独立同分布的训练样本 $Tr = \{(\boldsymbol{x}^{[1]}, y^{[1]}), \ldots\ldots, (\boldsymbol{x}^{[N]}, y^{[N]})\}$，深度神经网络的参数优化旨在找到一个使损失函数 $L(\boldsymbol{x}^{[i]}, y^{[i]}; \boldsymbol{w})$ 最小化的模型。

$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) = L(\boldsymbol{x}^{[1]}, y^{[1]}; \boldsymbol{w}) + \cdots + L(\boldsymbol{x}^{[N]}, y^{[N]}; \boldsymbol{w}). \tag{2.14}$$

一阶优化方法，也称为基于梯度的优化，基于一阶导数。其要求是损失函数 $L(\boldsymbol{w})$ 平滑，即连续，并且在几乎所有参数值 $\boldsymbol{w} = (w_1, \ldots, w_k)$ 处可微。这样，几乎可以在所有点计算 $L(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 的任意分量 w_j 的偏导数 $\frac{\partial L(\boldsymbol{w})}{\partial w_j}$。$L(\boldsymbol{w})$ 在特定点 $\boldsymbol{w}$ 处的梯度 是向量

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = \left(\frac{\partial L(\boldsymbol{w})}{\partial w_1}, \ldots, \frac{\partial L(\boldsymbol{w})}{\partial w_k} \right)^{\mathsf{T}}. \tag{2.15}$$

梯度指向 $L(\boldsymbol{w})$在点 $\boldsymbol{w}$ 处具有最陡上升的方向。因此，最速下降的方向与之相反$-\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}}$。因此，批量梯度下降算法会沿负梯度方向改变当前参数$\boldsymbol{w}_{(t)}$，以更接近最小值

$$\boldsymbol{w}_{(t+1)} = \boldsymbol{w}_{(t)} - \lambda \frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}}. \tag{2.16}$$

学习率 λ 决定步长，即在每次迭代中移动的幅度，直到达到最优值。由于每个参数$\boldsymbol{w}_{(t)}$ 的梯度通常都不同，因此必须针对每个新的参数向量重新计算梯度（图 2.19）。迭代过程会持续进行，直到导数接近于零。梯度为零表示达到了局部最小值或鞍点 [51, 第 79] 页。在实际应用中，只需从不同的 $\boldsymbol{w}$值开始重复优化，并在导数接近于零时停止即可。

深度神经网络通常需要数百万个训练样本。对所有这些样本反复计算梯度的成本极高。**随机梯度下降**（*SGD*）算法不使用整个数据集，而是每次只针对一个小小批量 的 m 训练样本计算梯度。通常，小批量的大小为 m 32 至 1024；对于近期出现的超大型模型，这一数值甚至更高。随后，根据（2.16）更新模型参数。

每次迭代都会从训练数据中随机选取一个新的小批量。根据大数定律，由这些小批量计算得出的梯度会围绕整个训练集的真实梯度波动。因此，平均而言，小批量梯度能够为参数更新指示恰当的方向。Mertikopoulos 等人 [91] 表明，通过迭代将学习率逐渐降至 0，SGD 几乎必然收敛，能够避开鞍点等伪临界点（概率为 1），并迅速稳定在局部极小值处。SGD 算法有多种变体，以下将对此进行介绍[65]。

优化中的一个重要步骤是参数初始化。参数的初始值会影响算法能否收敛，以及优化逼近最优解的速度。为打破对称性，初始参数必须是随机的。此外，应设置每一层参数的均值和方差，使该层输出呈现良好的分布，例如期望值为 0.0、方差为 1.0。此外，所有梯度也应呈现这种良性分布，以避免梯度爆炸或梯度消失。所有深度学习软件框架都包含适当的初始化程序。Goodfellow 等人对此作了详尽介绍。 [51,第 292] 页。

随机梯度下降的变体

动量 是一种帮助 SGD 沿相关方向提高收敛速度并减少振荡的方法。基本上，会计算近期梯度的移动平均值 $\boldsymbol{u}_{(t)}$ ，其中参数为 $\gamma \approx 0$。9，然后使用该平均值执行参数更新，方法是

$$\boldsymbol{u}_{(t)} = \gamma \boldsymbol{u}_{(t-1)} - \lambda \frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} \qquad \text{where} \qquad \boldsymbol{w}_{(t)} = \boldsymbol{w}_{(t-1)} - \boldsymbol{u}_{(t)}. \tag{2.17}$$

请注意，除了参数向量 $\boldsymbol{w}_{(t)}$ 外，还必须存储长度相同的移动平均值 $\boldsymbol{u}_{(t)}$，这需要与参数向量 $\boldsymbol{w}$ 相同的内存空间。这

在参数数量接近数十亿时，可能会消耗大量额外内存。近年来，又开发出了许多其他优化器 [65]:

- **AdaGrad** 会根据先前的梯度动态调整学习率。对于频繁出现的特征，它采用较小的学习率；对于很少出现的特征，则采用较大的学习率。
- **AdaDelta** 对 AdaGrad 进行了改进。它不再累积所有过去的梯度，而是将过去梯度的累积窗口限制为某个固定大小k。
- **RMSProp** 也是一种针对每个参数调整学习率的方法。其思路是，将某个权重的学习率除以该权重近期梯度幅值的滑动平均值。
- **Adam** 结合了 AdaGrad 和 RMSProp 各自的优点。Adam 基于低阶矩的自适应估计，同时对梯度及其二阶矩进行滑动平均。

由于预训练语言模型的参数数量极其庞大，共轭梯度 *ConjugateGradient* 或拟牛顿法 *Quasi-Newton* 等二阶优化方法很少使用。由于二阶导数的数量呈二次增长，因此只能采用粗略的近似方法。前文所述的 Adam 就是一个例子。

为了改进训练，预训练语言模型（PLM）的一项重要架构改进是残差连接，Vaswani 等人 [141]为 Transformer 提出了这一设计。事实证明，残差连接对于 ResNet [54] 等图像分类网络非常成功，并使训练数百层的网络成为可能。恒等捷径连接会跳过若干层组成的模块，以保留特征。Zhang 等人 [163]分析了包含残差连接的网络的表示能力。

大型模型并行训练

最近，有人建议采用更大的小批次来减少优化工作量。You 等人 [159] 提出了**LAMB优化器**，它采用逐层自适应学习率，以加速使用大批次训练预训练语言模型。他们证明，在一般的非凸情形下，该方法能够收敛到一个驻点。其实证结果显示，LAMB 的性能更优。只需进行极少的超参数调优，使用 32,868 的批量大小，就能将 BERT 的训练时间从 3 天缩短至仅 76 分钟，而且性能不会有任何下降。LAMB 程序代码可在线获取 [97]。此外，还可以降低优化过程的内存需求 [119]，从而实现模型并行化并提高训练速度。

大型模型（如 GPT-3）拥有数十亿个参数，无法再装入单个计算设备（例如 GPU）的内存中。因此，计算必须分配到多个 GPU 上。并行化技术有多种 [156]:

- 数据并行会将相同的模型代码和参数分配给每个 GPU，但使用不同的训练样本 [72]。梯度会并行计算，最后再汇总。
- 流水线并行 会将模型划分为不同部分（例如各层），并在不同的 GPU 上执行。如果某个部分计算完成，就会将结果发送到下一块 GPU。在训练的反向传播过程中，这一序列会反向进行。
- 层内模型并行 会将单个层的权重分布到多块 GPU 上。

为模型实现并行化策略是一个繁琐的过程。**DeepSpeed** 库 [122]提供了支持，使分布式训练变得简单、高效且有效。近期开发的 **GSPMD**系统 [156] 能够自动完成这一过程，并以统一的方式组合不同的并行范式。GSPMD 根据有限的用户注释推断模型定义中计算的分布方式，并将计算分配到 GPU 网络上。例如，它曾用于将拥有 1 万亿个参数的模型分布到 2,048 块 GPU 上。

2.4.2 预训练语言模型正则化

如果一个模型包含过多参数，那么它可以通过优化几乎完美地适应训练数据，反映训练数据中的几乎所有细节。在这种过拟合过程中，模型会学习训练数据中体现的随机变化，并偏离底层分布的均值。因此，它在测试数据上的性能通常较低，且具有更大的泛化误差。为避免这一现象，必须通过正则化方法降低模型的表征能力，这些方法通常与减少参数数量具有相同的效果。深度学习模型中广为人知的方法包括 L_2 正则化和L_1 正则化，它们会惩罚较大的参数值；或者使用*Dropout* ，将随机选定的隐变量暂时设为 0。 [96]对深度神经网络的正则化策略进行了综述。

PLM 的训练通常并非易事。其中一个问题是梯度消失或梯度爆炸的出现，这与不同层输入值的方差消失或爆炸问题相关 [55]。批归一化将隐藏单元各组件的值归一化为均值 0.0、方差 1.0，从而减少输入值的变化。对于一个训练样本小批量，会聚合各组件的值以计算均值和方差，然后使用这些均值和方差对每个训练样本中该组件的输入进行归一化 [62]。可以证明，批归一化会使深度神经网络各层之间的隐藏表示变得越来越正交 [35]。

在关于 Transformer 的论文中，Vaswani 等人 [141] 使用了一种称为层归一化 [6]的变体进行正则化。作者针对每个训练样本计算隐藏单元中不同组件的均值和方差，并据此将输入归一化，使均值为 0.0、方差为 1.0。此外，他们对自注意力的输出应用 Dropout。最后，他们使用标签平滑 [133] ，其中损失

函数经过重新表述，使得观测到的词元并非确定无疑，而是其他词元也可能以较小的概率出现。这是一种正则化形式，能让优化更容易。RMSNorm [162] 是层归一化的一种变体，它仅通过除以均方根误差来归一化输入，而不对均值进行平移。在实验中，它的表现优于层归一化 [101]。

2.4.3 神经架构搜索

自注意力模块的结构是手动设计的，目前尚不清楚它在所有情况下是否都是最优的。因此，有一些方法可以自动生成预训练语言模型（PLM）的架构，称为神经架构搜索（*NAS*）。He 等人 [56], 对此进行了综述，他们认为目前架构搜索对自然语言处理任务的贡献较小。Zöller 等人 [166] 评估了机器学习模型的架构搜索。

Wang 等人 [149] 提出了一种架构搜索空间，其中包含灵活的编码器—解码器注意力和异构层。架构搜索会生成多个 Transformer 版本，最终侧重于硬件限制，以便使计算适配手头的处理器。作者报告称，在性能没有损失的情况下，速度提升了 3 倍，模型大小缩减至原来的 1/3.7。对于关系分类，Zhu 等人 [165] 设计了一个全面的搜索空间。他们采用强化学习策略探索该搜索空间，并得到了性能更好的模型。

架构搜索也可以表述为排序任务。**RankNAS** [60]通过一系列二元分类问题来解决这一任务。作者研究了翻译和语言模型。对于翻译，通常的编码器-解码器被纳入一个超网络，其中每个 10^{23} 子网络都是一种独特的架构。架构特征（例如层数）的重要性，通过置换该特征后模型误差的增加来衡量。作者采用进化优化策略，并在翻译任务（WMT2014 英德翻译任务）上评估其方法。与其他方法相比，他们以更低的成本提升了 BLEU 值。

近年来，可微架构搜索逐渐发展起来。它将架构搜索嵌入连续的搜索空间，并通过梯度下降找到最优架构。这使搜索过程的效率大幅提升，速度比离散搜索方法快几个数量级。 [43], 将这一思路应用于机器翻译，提出了一种基于梯度的 NAS 算法。他们探索了注意力模块和循环单元，并自动发现性能更优的架构。不同单元之间的连接拓扑以端到端的方式学习。在多个基准测试中，他们提升了 Transformer 的性能；例如，在 WMT2014 英德翻译任务中，BLEU 分数从 28.8 提高到了 30.1。其他成功的神经机器翻译架构搜索方法 [130],、命名实体识别方法 [64], 以及图像分类模型 [34, 147, 148], ，也可能应用于其他自然语言处理任务。

2.4.4 模型预测不确定性

预训练语言模型（PLM）的结果差异主要有两个来源：

- 认知不确定性反映了我们对现实世界的认知有限。与训练集相对应的现实世界情况可能发生变化，从而导致分布偏移。此外，收集到的文档可能存在偏见或错误，也可能包含不需要的内容类型。显然，现实世界的结构与预训练语言模型的结构并不相同。因此，预训练语言模型只能近似估计语言的正确条件概率。这种类型的不确定性通常称为结构不确定性，且难以估计。

- 偶然不确定性由随机变化引起，较容易评估。训练数据通常是总体中底层数据的一个样本，因此会受到抽样变异的影响。如果对模型进行随机重新初始化，它会生成一组完全不同的参数值，从而导致不同的预测。最后，语言模型会预测词元的概率，而新词元的生成也会受到不确定性的影响。贝叶斯框架为评估深度学习中的此类不确定性提供了一种理论依据充分的工具 [44]。

Gaw-likowski 等人最近提供了一篇关于估计模型不确定性方法的综述 [47]。我们将介绍三种捕捉模型不确定性的方法：贝叶斯统计、狄利克雷分布和集成分布。

贝叶斯神经网络

贝叶斯神经网络 通过后验分布直接表示所估计参数 $\boldsymbol{w} = (w_1, \ldots, w_{d_w})$的不确定性

$$p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y}) \propto p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w})p(\boldsymbol{w}). \tag{2.18}$$

其中， $\boldsymbol{X}$和 $\boldsymbol{Y}$是训练集中的观测输入和输出，而 $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{w})$ 是似然，即在给定 $\boldsymbol{X}$ 和参数向量$\boldsymbol{w}$时输出的概率。先验分布 $p(\boldsymbol{w})$描述了数据可用之前参数的分布。对于新输入 $\tilde{\boldsymbol{x}}$ ，预测的分布由下式给出

$$p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}}, \boldsymbol{X}, \boldsymbol{Y}) = \int p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}}, \boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})d\boldsymbol{w}. \tag{2.19}$$

该积分通常无法解析求解，必须进行近似。通常使用蒙特卡洛 近似，即通过对根据后验分布 $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})$分布的不同参数值 $\boldsymbol{w}^{[i]}$ 求和来推断该积分。如果 $\tilde{\boldsymbol{y}}^{[i]} = f(\tilde{\boldsymbol{x}}, \boldsymbol{w}^{[i]})$ 是一个确定性网络，用于针对参数$\boldsymbol{w}^{[i]}$ 和输入 $\tilde{\boldsymbol{x}}$预测输出，那么所得样本 $\tilde{\boldsymbol{y}}^{[1]}, \ldots, \tilde{\boldsymbol{y}}^{[k]}$ 可以被视为输出分布 $p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}}, \boldsymbol{X}, \boldsymbol{Y})$ [108]的一个样本。

贝叶斯预测分布可以通过不同方式近似：

- 采样方法 使用马尔可夫链蒙特卡罗 算法生成服从后验分布的参数值，并据此进行采样 [102]。马尔可夫链蒙特卡罗定义了一种采样策略：首先随机生成一个新的参数值 $\boldsymbol{w}$，然后算法计算接受该值的概率 $\boldsymbol{w}$，或保留之前参数值的概率。Welling 等人 [150] 将这种方法与随机梯度下降相结合，并证明可以通过带噪 SGD 对深度神经网络进行贝叶斯推断。Nemeth 等人 [103]综述了其良好的收敛性质。Wenzel 等人 [152]对这项技术进行了实际评估。

- 变分推断 通过较简单分布的乘积 $q(\boldsymbol{w})$ 来近似后验分布，这些分布更容易求值 [9]。借助多块 GPU 以及数据增强、动量初始化和学习率调度等实用技巧，Osawa 等人 [105] 证明，变分推断可以扩展到 ImageNet 规模的数据集和架构。

  可以证明 [45]，丢弃正则化（第 2.4.2 节）可视为近似变分推断。因此，估计预测不确定性时，不仅可以在训练期间使用 Dropout，也可以在测试时使用。一个名为*DropConnect* 的变体会随机移除某个节点的输入激活，而不是丢弃所有后续节点共享的一个激活。这种方法可以得到更可靠的不确定性估计，甚至可以与原始 Dropout 技术结合使用 [88]。

- 拉普拉斯近似 考察局部模式 $\hat{\boldsymbol{w}}$ 附近后验分布的对数，并用网络权重 [9] 上的正态分布 $N(\hat{\boldsymbol{w}}, [H+\beta I]^{-1})$ 进行近似。H 是 $\log\ p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{Y})$ 的海森矩阵，即二阶导数矩阵。这种近似可以针对已训练好的网络计算，并可应用于深度神经网络 [76]。一个问题是 H的系数数量庞大，这使计算仅限于对角线上的元素。 [48]提出了若干扩展方法。

通过单个确定性模型估计不确定性

大多数预训练语言模型通过离散概率分布预测词元。如果使用 softmax 函数计算这些概率，对训练集进行优化通常会导致非常极端的概率值，接近 0 或 1。网络往往过度自信，并生成不准确的不确定性估计。为了评估不确定性，必须描述估计分布与实际分布之间的差异。如果 v_1、.....、 v_{d_v} 是词元的词汇表，$\boldsymbol{\pi}$ 是这些词元上的离散分布，那么我们可以使用 狄利克雷分布 $p(\boldsymbol{\pi}|\boldsymbol{\alpha}(\boldsymbol{x}))$ 来刻画这些离散分布上的分布。向量 $\boldsymbol{\alpha}$ 取决于输入 $\boldsymbol{x}$，并且为每个 v_i包含一个分量 α_i。总和 $\sum_i \alpha_i$ 刻画了方差。如果该总和变大，则 v_i 的概率估计具有更低的方差。

Malinin 等人 [85]利用经验分布与预测分布之间的预期散度，估计给定输入 $\boldsymbol{x}$ 的 $p(\boldsymbol{\pi}|\boldsymbol{\alpha}(\boldsymbol{x}))$。在

训练数据所在区域内，网络经过训练以最小化分布内数据的预测与低方差狄利克雷分布之间的预期*Kullback-Leibler* (*KL*)散度。在分布外数据所在区域内，则估计方差较高的狄利克雷分布。输出上的分布可解释为对模型不确定性的量化，旨在模拟对网络参数 [44] 进行贝叶斯建模的行为。

Liu 等人 [83]认为，训练数据元素之间的距离与预测不确定性有关。为避免网络中的层导致输入空间距离发生严重失真，作者提出了谱归一化。这种**SNGP**方法将距离 $||h(\boldsymbol{x}^{[1]}) - h(\boldsymbol{x}^{[2]})||$限制在相对于 $||\boldsymbol{x}^{[1]} - \boldsymbol{x}^{[2]}||$的范围内，其中 $\boldsymbol{x}^{[1]}$ 和 $\boldsymbol{x}^{[2]}$ 是两个输入，$h(\boldsymbol{x})$ 是一个深度特征提取器。随后，他们将$h(\boldsymbol{x})$传入一个具有距离感知能力的高斯过程 输出层。高斯过程后验通过拉普拉斯近似进行近似，并可由确定性的深度神经网络进行预测。

作者在 $\text{BERT}_{\text{BASE}}$ 上评估 SNGP，以判断自然话语输入是否包含在训练数据中（从而可以由模型处理），或是否属于训练数据之外。模型仅使用域内数据进行训练，并在域内数据和域外数据上评估其预测准确率。虽然集成技术具有略高的预测准确率，但 SNGP 在概率校准和分布外检测方面表现更好。该方法的实现可在 [138]处获取。

第 [47, 10 页及后续页], 介绍了若干替代方法，其中也讨论了如何利用狄利克雷分布混合来刻画预测不确定性。一般而言，与其他方法相比，单一确定性方法在训练和评估时的计算开销较低。然而，这类方法依赖单一网络配置，可能对底层网络结构和训练数据非常敏感。

用集成表示预测分布

可以通过重抽样方法来模拟训练集的抽样变异性。一种理论基础扎实的方法是装袋法：从包含 n 个元素的训练集中有放回地抽取 n 个样本，得到 n_b 个样本 [20, 107]。对于第 i 个样本，可以训练一个模型，得到参数 $\hat{\boldsymbol{w}}^{[i]}$。随后，预测分布 $f(\boldsymbol{x}, \hat{\boldsymbol{w}}^{[i]})$ 表示模型对输入 $\boldsymbol{x}$ 的预测不确定性；可以证明，其均值 $\frac{1}{n_b}\sum_i f(\boldsymbol{x}, \hat{\boldsymbol{w}}^{[i]})$ 的方差低于原始模型的预测 [73]。与许多近似方法不同，集成方法可以考虑似然函数的不同局部极大值，并且可能涵盖不同的网络架构。还有其他引入数据变化的方法，例如随机参数初始化或随机数据增强。 [40] 提供了集成方法综述。

除了提高准确率之外，集成模型还被广泛用于表示深度神经网络的预测不确定性 [73]。在实证研究中，该方法至少与贝叶斯方法（蒙特卡洛丢弃、概率反向传播）同样可靠 [73]。重新排列训练数据并随机初始化参数，会使模型产生足够的变异性，从而能够预测不确定性，而随机

参数初始化会使模型产生足够的变异性，从而能够预测不确定性，而装袋法可能会降低不确定性估计的可靠性[77]。与蒙特卡洛丢弃相比，集成模型产生的预测不确定性更可靠、校准效果更好，并且适用于真实世界的训练数据[13,53]。即使集成规模相对较小、仅为五个，深度集成似乎也表现最佳，而且比所比较的方法更能适应数据集偏移[106]。

尽管预训练语言模型已被用作大多数自然语言处理任务的标准解决方案，但现有模型中的大多数无法估计与其预测相关的不确定性。这似乎主要是由不确定性估计方法的高计算开销所致。此外，预测概率分布的不确定性这一概念难以传达。然而，当向预训练语言模型提供超出其训练数据支撑范围的输入时，获得诊断结果极其重要，因为此时预测会变得不可靠。

在所讨论的方法中，集成方法似乎最为可靠。然而，它们需要极高的计算开销。像 SNGP 这样的新算法非常有前景。仍需更多研究来降低这一开销或开发替代性方法。近年来，已开发出基准测试代码库和数据集，以提供标准方法和最先进方法的高质量实现，并描述不确定性和鲁棒性基准测试的最佳实践 [99]。

实现:

不确定性基准 [10, 98]提供了一系列用于不确定性评估的标准方法和最先进水平方法的高质量实现。

2.4.5 模型预测解释

诸如 BERT 之类的预训练语言模型被视为黑箱模型，因为人们很难理解它们究竟学到了什么，以及哪些因素决定了它们的输出。因此，大量研究致力于探究这些模型的行为。解释模型预测主要有三个原因。信任 模型预测是必要的，即模型能够针对当前问题生成可靠的答案，并可部署于现实世界的应用中。因果性 断言，输入属性的变化会导致模型预测发生合理的变化。理解 模型有助于领域专家将模型预测与现有领域知识进行比较。这是通过融入领域知识来调整预测模型能力的先决条件。

解释还可用于调试模型。一个引人注目的示例是图像分类：模型并不是根据形状检测到马，而是根据图像中的一个标签检测到马[74]。对于涉及人类或可能造成重大损害的关键决策，解释尤为重要。例如医疗保健、司法系统、银行业或自动驾驶汽车。

解释方法大致可以分为局部解释或全局解释。局部解释为特定输入 $\boldsymbol{x}$ 的模型预测提供信息或依据，而全局解释则涵盖模型的整体情况

。绝大多数模型旨在提供局部解释，因为这些解释可用于证明特定预测的合理性。 [11, 12, 23, 36, 139, 155] 对预训练语言模型解释方法的综述进行了介绍。Molnar [95] 用整本书探讨了这一主题，Bommasani 等人 [17，第 125] 页提供了最新概述。对于语言模型，可以使用不同类型的解释：

- **特征重要性**衡量单个输入特征（例如词元）对预测的影响。它通常对应于特征相对于输出 [79] 的一阶导数。由于输入词元的意义很容易理解，因此人类很容易理解这类解释。
- **反事实解释**探究如何修改输入 $\boldsymbol{x}$，才能生成不同的目标输出。
- **代理模型**通过另一个更简单的模型来解释模型预测。一个著名的示例是 *LIME* [123], ，它围绕单个感兴趣的输入 $\boldsymbol{x}$ 训练局部线性模型。
- **示例驱动的**解释通过选择其他语义上与 $\boldsymbol{x}$ 相似的带标签实例，来说明输入 $\boldsymbol{x}$ 的预测。这与用于预测的最近邻方法相近，例如已用于文本分类 [1]。
- **来源引用**是科学研究中的一种普遍做法，即通过引用可靠的科学来源来支撑一项主张。对于带有检索组件的语言模型生成的文本，也可以采用同样的做法 [57]。

其他方法，例如由一系列推理步骤或规则调用组成的方法，并不适用于拥有数百万个参数的预训练语言模型。

自注意力机制是预训练语言模型的核心功能单元。**BertViz** [144]是一种可视化工具，允许用户探索预训练语言模型中不同词元之间的注意力强度（按注意力头和层划分），并快速概览相关的注意力头。然而，Jain 等人 [63] 证明，注意力与特征重要性方法并无相关性，而对注意力进行反事实变化也不会导致预测发生相应变化。这可能是因为注意力头输出经过拼接后，随后由全连接非线性层进行处理。注意力是衡量组件整体重要性的嘈杂预测指标，但并不善于识别特征的重要性 [129]。

线性局部近似

一个重要的概念是输入 x_i 对输出 y_j 的贡献，例如类别概率。基于梯度的解释通过计算偏导数 $\partial y_j / \partial x_i$，估算输入 x_i 对输出 y_j 的贡献，例如类别概率。这个导数通常称为显著性，可解释为输入 $\boldsymbol{x}$ 处预测函数的线性近似。**LIME** [123] 围绕单个输入 $\boldsymbol{x}$ 定义了一个局部线性回归模型。由于特征之间存在相关性，输入特征的系数取决于其他特征的有无



图 2.20 问题分类任务的归因（左）。红色表示正面影响，蓝色表示负面影响，黑色词元表示中性。右侧展示了将“早上好，女士们、先生们”翻译成德语“*GutenMorgen Damenund Herren*”这一任务的归因 [132]。单词被分词为词片段。

输入特征。**SHAP** 方法因此通过计算某个特征在其他特征的所有组合 [84]下的平均影响，来确定该特征的影响。作者展示了这种方法良好的理论性质，并推导出几种高效的计算策略。

非线性局部近似

Sundararajan 等人 [132] 提出了对此类解释的两项基本要求。敏感性：如果输入 $\boldsymbol{x}^{[1]}$ 和 $\boldsymbol{x}^{[2]}$ 仅在一个特征上存在差异，并导致不同的预测，那么该差异特征应获得非零归因。实现不变性：也就是说，对于两个功能等价的网络，其归因始终相同。由于预测函数通常是非线性的，基于梯度的方法会违反这两项要求，并且可能聚焦于无关属性。

集成梯度 [132] 生成了预测函数 $F: \mathbb{R}^n \to [0,1]$ 的近似值，该函数捕捉非线性依赖关系。为了评估基线输入 $\boldsymbol{x}^{[1]}$ 与另一个输入 $\boldsymbol{x}^{[2]}$之间的差异，作者通过积分计算从 $\boldsymbol{x}^{[1]}$ 到 $\boldsymbol{x}^{[2]}$ 的直线上，输出相对于输入的梯度 $\partial F(\boldsymbol{x})/\partial \boldsymbol{x}$ 的均值。可以证明，这种方法满足上述要求。作者将该方法用于按回答类型进行问题分类（图 2.20）。基线输入是全零嵌入向量。另一项应用涉及神经机器翻译。在此，每个输出词元的输出概率都归因于输入词元。作为基线，除起始和结束标记外，所有词元都被置零。另一种类似的分析基于预测函数 [7] 的泰勒展开。

Liu 等人 [82]提出了一种生成式解释框架，能够同时学习做出分类决策并为其生成细粒度解释。为了在分类与解释之间建立良好的联系，他们引入了一个基于解释进行训练的分类器。对于产品评论，他们

例如，生成以下正向解释 “画面出色，玻璃背板屏幕引人注目，支持 *HDR10* 和 *Dolby Vision*” 以及负面理由 “价格非常昂贵”。作者引入了一种解释因子，用来表示基于解释训练的分类器与基于原始输入和金标准标签训练的分类器之间的概率差异。他们使用最小风险训练来优化模型。

基于检索的解释

近年来，深度学习模型在科学和科技领域发挥着日益重要的作用。Facebook 开发的算法在预测用户偏好方面比任何心理学家都更加出色 [24, 71]。DeepMind 开发的 AlphaFold 根据氨基酸预测蛋白质结构，其准确率最高 [131]。PaLM 和 Retro 模型能够以流畅的英语生成故事，后者以互联网中的知识作为背景。然而，这些程序实际上都无法为其决策提供依据，也无法说明为何生成了某个特定序列，或某项决策是基于哪些信息作出的。

2008 年，Anderson [5] 预言了基于理论的科学的终结。在他看来，理论是对现实的过度简化，而大量积累的数据以更加详细的形式包含了知识，因此理论已不再必要。这同样是可解释人工智能面临的问题，它旨在解释深度学习模型的决策。它始终面临着一种权衡：为了理解模型输出，必须牺牲预测准确率。

随着大型自回归语言模型与检索组件相结合，文档检索不仅可用于在语言生成过程中融入更准确的知识，还能通过权威引文为生成的答案提供支撑。Metzler 等人 [92] 认为，未来的预训练语言模型应通过引用训练数据或背景文档集中的支持性文档，为生成的文本提供依据。为了实现这一方法，Nakano 等人 [100] 将*GPT-3*与搜索引擎必应 相结合，通过检索到的文档增强问答任务中的语言生成。其**WebGPT** [100]首先生成一段自然语言文本（第 6.2.3 节）。随后，它通过引用找到的文档来补充生成的句子，这与科学家通过引用文献来扩展文本的方式相似。通过这一过程，WebGPT 能够为生成的答案提供依据并加以解释。这或许能让生成的文本更值得信赖。请注意，先进的对话模型**LaMDA** 能够附上支持回答的外部文档链接（第 6.6.3 节）。

通过生成思维链进行解释

像 GPT-3 这样的大型自回归预训练语言模型能够生成非常令人信服的起始文本续写，例如生成问题的回答。事实证明

思维链示例：我将解释如何进行逻辑推理：
(1) John 被孩子们在门外的咯咯笑声吵醒。他们冲进来，然后
端着满满一盘食物，说道：“父亲节快乐！”2022年是美好的一年。我们能否
推断出 John 出生于巴拉克·奥巴马就职典礼之前?
回答：我们可以从“他的孩子们”推断出 John 是一位父亲。我们可以推断出
现在是2022年。巴拉克·奥巴马于2009年就职，因此距今大约
奥巴马就职已过去13年。John的孩子们已经到了会说话的年龄，因此如果
John出生在奥巴马就职之后，那么在他有孩子时，他还不到10岁。
他有了孩子。因此，我们可以推断John出生在奥巴马就职之前。
回答是“是”。
(2) Sam看到一架钢琴.....
输入查询：Jennifer 从窗户向外看，看到她下方有一朵非常酷的云。她
解开安全带，前往洗手间。Jennifer 可能正在以更快的速度行驶吗
相对于地球而言超过每小时 300 英里?

模型输出：每小时 300 英里约等于每小时 480 千米。这大约是一架
商用飞机的速度。云通常位于飞机下方，因此 Jennifer 可能正在乘坐一架
飞机。回答是“是”。

图 2.21 通过思维链进行解释。第一个方框包含两个思维链示例，这些示例用于每个查询。该思维链提示与输入查询一起输入 PaLM 模型，并由 PaLM [30, p. 38]生成模型输出。

只需提供几个带有思维链（第 3.6.4 节）的示例，展示如何推导出正确答案，就能大幅提高它们生成正确答案的能力。PaLM语言模型 [30] 已经展示了这一点。

生成的思维链 还可用于其他目的。首先，可以检查模型是否基于“正确的理由”给出正确答案，而不是仅仅利用表面统计相关性。此外，可以向系统的终端用户展示解释，从而增强或降低他们对某个预测的信心。最后，对于某些查询（例如解释一个笑话），解释本身就是所需的输出 [30]。

图 2.21 展示了一个少样本查询及其回答。在实际应用中，只需少量思维链示例，并且可以重复使用。为了针对问题生成最佳回答，必须使用贪心解码，以得到最优预测。PaLM 的结果表明，列举论证步骤在经验上是有效的。然而，目前仍缺乏关于模型实际上如何在内部使用此类论证的可靠理论。此外，尚不清楚在什么情况下能够成功推导出这样的思维链。还应研究模型的推理在多大程度上与人类执行的推理步骤相对应。

实现

Ecco [2] 和BertViz [143] 是用于可视化预训练语言模型的注意力和嵌入的工具。TensorFlow [136]提供了集成梯度的实现和教程。Captum [26, 70] 是一个开源库，可为 PyTorch 模型的预测生成解释和说明，其中包含上文讨论的大多数方法

Transformers-interpret [113]是 Hugging Face 软件包的另一款开源模型可解释性工具。

2.4.6 总结

与其他大型神经网络类似，预训练语言模型采用简单的随机梯度下降优化器进行优化，即使对于拥有数十亿个参数和数太字节训练数据的超大型模型，这些优化器也能使模型接近成本最低的区域。这需要在计算网络上进行并行训练，并通过合适的软件库加以控制。文献中有许多关于如何设置批量大小、学习率调度等超参数的经验方法。重要的组成要素包括残差连接，以便优化层数众多的网络；以及正则化模块，以使参数保持在可控范围内。

神经架构搜索是一种提升网络性能并降低其内存需求的方法。已有多种方法被提出，可显著加快训练速度。一些方法能够构建性能更好、内存占用更低的模型。还有一些新的微分方法，有望以较小的工作量推导出更好的架构。

预训练语言模型旨在捕捉语言概念之间的关系，但只能近似实现这一目标。因此，评估其固有的不确定性十分重要。文中介绍了三种分析不确定性的方法。其中，集成方法似乎最可靠，但计算成本较高。诸如 SNGP 这类基于单一模型的新算法前景十分可观。

为了让用户能够判断模型结果是否合理，有必要解释结果是如何得出的。可以通过展示各项特征对结果的重要性、利用相关示例探查预训练语言模型，或用简单模型近似预训练语言模型来提供解释。已有一些库可供日常使用这些方法。解释预训练语言模型生成文本的一种新方法，是为文本添加相关支持性文档的适当引文。最后，可以通过思维链提示指示预训练语言模型解释模型的响应。这种类型的解释尤其容易理解，也能反映一系列论证中的关键部分。

下一章将探讨通过引入新的预训练任务或进行架构变更来改进三种基本预训练语言模型类型的方法。第四章将考察预训练语言模型能够获取的知识，以及如何利用这些知识来解读文本并生成新文本。

参考文献

[1] A. Abujabal、R. S. Roy、M. Yahya 和 G. Weikum。“Quint：基于知识库的可解释问答”。载于：2017年自然语言处理经验方法会议论文集及系统演示，2017年，第61–66页。[2] J. Alammar。“Ecco：用于解释 Transformer 语言模型的开源库”。载于：第59届计算语言学协会年会暨第11届自然语言处理国际联合会议系统演示论文集，2021年，第249–257页。网址：https://github.com/jalammar/ecco。[3] J. Alammar。《图解 GPT-2》（可视化 Transformer 语言模型）。2019年10月。网址：http://jalammar.github.io/illustrated-gpt/（访问日期：2021年1月24日）。[4] F. St-Amant。《如何微调 GPT-2 以生成文本》。Medium。2021年5月8日。网址：
htt
ps://towardsdatascience.com/how-to-fine-tune-gpt-2-for-text-generation-ae2ea53bc272（访问日期：2021年7月29日）。[5] C. Anderson。“理论的终结：数据洪流使科学方法过时”。载于：《Wired》（2008年6月23日）。ISSN：1059-1028。网址：https://www.wired.com/2008/06/pb-theory/（访问日期：2022年1月11日）。[6] J. L. Ba、J. R. Kiros 和 G. E. Hinton。“层归一化”。2016年。arXiv：1607.06450。[7] S. Bach、A. Binder、G. Montavon、F. Klauschen、K.-R. Müller 和 W. Samek。“通过逐层相关性传播解释非线性分类器决策的逐像素解释”。载于：《PloS one》10.7（2015），e0130140。[8] D. Bahdanau、K. Cho 和 Y. Bengio。“通过联合学习对齐与翻译实现神经机器翻译”。2014年。arXiv：1409.0473。[9] D. Barber 和 C. M. Bishop。“贝叶斯神经网络中的集成学习”。载于：Nato ASI Ser. F Comput. Syst. Sci. 168（1998），第215–238页。[10] baselines。《不确定性基准》。Google，2021年12月5日。网址：https://github.com/google/uncertainty-baselines（访问日期：2021年12月6日）。[11] C. Bauckhage、J. Fürnkranz 和 G. Paass。“可信、透明且稳健的机器学习”。载于：《人工智能手册》。de Gruyter，2021年。ISBN：978-3-11-065984-9。[12] V. Belle 和 I. Papantonis。“可解释机器学习的原则与实践”。载于：《Front. Big Data》4（2021），第39页。ISSN：2624-909X。DOI：10.3389/fdata.2021.688969。[13] W. H. Beluch、T. Genewein、A. Nürnberger 和 J. M. Köhler。“集成模型在图像分类主动学习中的力量”。载于：IEEE计算机视觉与模式识别会议论文集，2018年，第9368–9377页。[14] Y. Bengio、A. Courville 和 P. Vincent。“表示学习：综述与新视角”。载于：《IEEE模式分析与机器智能汇刊》35.8（2013），第1798–1828页。[15] Y. Bengio、R. Ducharme、P. Vincent 和 C. Jauvin。“神经概率语言模型”。载于：《机器学习研究杂志》3（2003年2月），第1137–1155页。[16] Y. Bengio、P. Lamblin、D. Popovici 和 H. Larochelle。“深度网络的贪婪逐层训练”。载于：《神经信息处理系统进展》19（2006）。[17] R. Bommasani 等人。“基础模型的机遇与风险”。2021年。arXiv：2108.07258。[18] S. Borgeaud 等人。“通过从万亿级词元中检索来改进语言模型”。2021年12月8日。arXiv：2112.04426 [cs]。[19] G. Branwen。“{GPT-2} 神经网络诗歌”。载于：（2019年3月3日）。网址：https://www.gwern.net/GPT-2（访问日期：2021年1月27日）。[20] L. Breiman。“装袋预测器”。载于：《机器学习》24.2（1996），第123–140页。[21] T. B. Brown 等人。“语言模型是少样本学习器”。2020年。arXiv：2005.14165。[22] D. Budden 和 M. F. Huber。《使用 JAX 加速我们的研究》。2020年12月4日。网址：https://www.deepmind.com/blog/using-jax-to-accelerate-our-research（访问日期：2022年6月21日）。[23] N. Burkart 和 M. F. Huber。“监督机器学习可解释性综述”。载于：《人工智能研究杂志》70（2021），第245–317页。

[24] C. Cadwalladr 和 E. Graham-Harrison。“剑桥分析如何将 Facebook 的‘点赞’变成有利可图的政治工具”。载于：《卫报》17032018（2018年）。[25] X. Cai、J. Huang、Y. Bian 和 K. Church。“上下文嵌入空间中的各向同性：簇与流形”。载于：国际学习表征会议，2020年。[26] Captum。Captum · PyTorch模型可解释性。2021年。网址：https://captum.ai/（访问于2021年12月6日）。[27] S. Chaudhari、V. Mithal、G. Polatkan 和 R. Ramanath。“注意力模型综述”。载于：ACM智能系统与技术汇刊，TIST 12.5（2021年），第1–32页。[28] S. F. Chen、D. Beeferman 和 R. Rosenfeld。“语言模型评估指标”。载于：（1998年）。网址：https://kilthub.cmu.edu/articles/Evaluation_Metrics_For_Language_Models/6605324/ files/12095765.pdf。[29] Y. Chen、V. O. Li、K. Cho 和 S. R. Bowman。“一种稳定有效的可训练贪心解码学习策略”。2018年。arXiv：1804.07915。[30] A. Chowdhery 等人。“PaLM：借助 Pathways 扩展语言建模”。2022年4月5日。arXiv：2204.02311 [cs]。[31] E. Cohen 和 C. Beck。“神经序列模型中束搜索性能退化的实证分析”。载于：国际机器学习会议。PMLR，2019年，第1290–1299页。[32] R. Collobert、J. Weston、L. Bottou、M. Karlen、K. Kavukcuoglu 和 P. Kuksa。“从零开始的自然语言处理（几乎）”。载于：《机器学习研究期刊》12（2011年），第2493–2537页。[33] A. M. Dai 和 Q. V. Le。“半监督序列学习”。载于：神经信息处理系统进展，2015年，第3079–3087页。[34] Z. Dai、H. Liu、Q. V. Le 和 M. Tan。“CoAtNet：融合卷积与注意力，适用于各种规模的数据”。2021年9月15日。arXiv：2106.04803 [cs]。[35] H. Daneshmand、A. Joudaki 和 F. Bach。“批归一化使深度随机网络中的表示正交化”。2021年6月7日。arXiv：2106.03970 [cs, stat]。[36] M. Danilevsky、K. Qian、R. Aharonov、Y. Katsis、B. Kawas 和 P. Sen。“自然语言处理领域可解释人工智能现状综述”。2020年。arXiv：2010.00711。[37] A. de Santana Correia 和 E. L. Colombini。“请注意！深度学习中的神经注意力模型综述”。载于：《人工智能评论》（2022年），第1–88页。[38] J. Devlin、M.-W. Chang、K. Lee 和 K. Toutanova。“Annotated BERT：用于语言理解的深度双向Transformer预训练”。载于：2019年计算语言学协会北美分会人类语言技术会议论文集，第1卷：长篇论文。NAACL-HLT 2019。美国明尼阿波利斯：计算语言学协会，2019年6月，第4171–4186页。doi：10.18653/v1/N19-1423。[39] J. Devlin、M.-W. Chang、K. Lee 和 K. Toutanova。“BERT：用于语言理解的深度双向Transformer预训练”。2018年。arXiv：1810.04805。[40] X. Dong、Z. Yu、W. Cao、Y. Shi 和 Q. Ma。“集成学习综述”。载于：《计算机科学前沿》14.2（2020年），第241–258页。[41] K. Doshi。《Transformer视觉解析（第3部分）：多头注意力深度解析》。Medium。2021年6月3日。网址：https://towardsdatascience.com/transformers-explained-visually-part-3-multi-head-attention-deep-dive-1c1ff1024853（访问于2021年11月19日）。[42] A. Fan、M. Lewis 和 Y. Dauphin。“层次化神经故事生成”。2018年。arXiv：1805.04833。[43] Y. Fan、F. Tian、Y. Xia、T. Qin、X.-Y. Li 和 T.-Y. Liu。“为神经机器翻译搜索更优架构”。载于：IEEE/ACM音频、语音与语言处理汇刊28（2020年），第1574–1585页。[44] Y. Gal 和 Z. Ghahramani。“采用伯努利近似变分推断的贝叶斯卷积神经网络”。2015年。arXiv：1506.02158。[45] Y. Gal、J. Hron 和 A. Kendall。“Concrete Dropout”。2017年。arXiv：1705.07832。[46] A. Galassi、M. Lippi 和 P. Torroni。“自然语言处理中的注意力”。载于：IEEE《神经网络与学习系统汇刊》32（2021年10月1日），第4291–4308页。doi：10.1109/TNNLS.2020.3019893。

[47] J. Gawlikowski 等人。“深度神经网络中的不确定性综述”。2021年。arXiv：2107.03342。[48] T. George、C. Laurent、X. Bouthillier、N. Ballas 和 P. Vincent。“在克罗内克分解特征基中快速近似自然梯度下降”。2018年。arXiv：1806.03884。[49] M. Geva、R. Schuster、J. Berant 和 O. Levy。“Transformer前馈层是键值记忆”。载于：（2020年12月29日）。网址：https://arxiv.org/abs/2012.14913v2（访问于2021年11月8日）。[50] B. Ghojogh 和 A. Ghodsi。“注意力机制、Transformer、BERT 与 GPT：教程与综述”。载于：（2020年）。网址：https://osf.io/m6gcn/download。[51] I. Goodfellow、Y. Bengio 和 A. Courville。《深度学习》。第1卷。剑桥：麻省理工学院出版社，2016年。网址：https://www.deeplearningbook.org/。[52] A. Graves。“使用循环神经网络进行序列转换”。2012年。arXiv：1211. 3711。[53] F. K. Gustafsson、M. Danelljan 和 T. B. Schon。“用于鲁棒计算机视觉的可扩展贝叶斯深度学习方法评估”。载于：IEEE/CVF计算机视觉与模式识别会议研讨会论文集。2020年，第318–319页。[54] K. He、X. Zhang、S. Ren 和 J. Sun。“用于图像识别的深度残差学习”。载于：IEEE计算机视觉与模式识别会议论文集。2016年，第770–778页。[55] K. He、X. Zhang、S. Ren 和 J. Sun。“深入研究整流器：超越人工水平的ImageNet分类性能”。载于：IEEE国际计算机视觉会议论文集。2015年，第1026–1034页。[56] X. He、K. Zhao 和 X. Chu。“AutoML：最先进水平综述”。载于：《知识型系统》212（2021年），第106622页。[57] J. Hilton。WebGPT：通过网页浏览提高语言模型的事实准确性。OpenAI。2021年12月16日。网址：https:// openai.com/ blog/improving- factual- accuracy/（访问于2022年1月12日）。[58] A. Holtzman、J. Buys、L. Du、M. Forbes 和 Y. Choi。“神经文本退化的奇特案例”。2020年2月14日。arXiv：1904.09751 [cs]。[59] J. Howard 和 S. Ruder。“用于文本分类的通用语言模型微调”。载于：计算语言学协会第56届年会论文集，第1卷：长篇论文。ACL 2018。澳大利亚墨尔本：计算语言学协会，2018年7月，第328–339页。doi：10.18653/v1/P18-1031。[60] C. Hu 等人。“RankNAS：通过成对排序实现高效神经架构搜索”。2021年。arXiv：2109.07383。[61] D. Hu。“NLP问题中注意力机制入门综述”。载于：SAI智能系统会议论文集。施普林格，2019年，第432–448页。[62] S. Ioffe 和 C. Szegedy。“批归一化：通过减少内部协变量偏移加速深度网络训练”。载于：国际机器学习会议。PMLR，2015年，第448–456页。[63] S. Jain 和 B. C. Wallace。“注意力并非解释”。2019年。arXiv：1902.10186。[64] Y. Jiang、C. Hu、T. Xiao、C. Zhang 和 J. Zhu。“面向语言建模和命名实体识别的改进型可微架构搜索”。载于：2019年自然语言处理实证方法会议暨第九届自然语言处理国际联合会议论文集。EMNLP-IJCNLP。2019年，第3576–3581页。[65] M. Kastrati 和 M. Biba。“机器学习高级优化方法的最先进水平综述”。载于：RTA-CSIT（2021年5月1日），第1–10页。[66] R. Kehlbeck、R. Sevastjanova、T. Spinner、T. Stähle 和 M. El-Assady。《揭开语言模型嵌入空间的奥秘》。2021年7月31日。网址：https://bert-vs-gpt2.dbvis.de/。[67] N. S. Keskar、B. McCann、L. R. Varshney、C. Xiong 和 R. Socher。“CTRL：用于可控生成的条件Transformer语言模型”。2019年9月20日。arXiv：1909. 05858。[68] U. Khandelwal、O. Levy、D. Jurafsky、L. Zettlemoyer 和 M. Lewis。“通过记忆实现泛化：最近邻语言模型”。2020年2月14日。arXiv：1911. 00172。

[69] D. P. Kingma 和 J. Ba。“Adam：一种随机优化方法”。2014年。arXiv：1412.6980。[70] N. Kokhlikyan 等人。“Captum：面向 PyTorch 的统一通用模型可解释性库”。2020年9月16日。arXiv：2009.07896。[71] M. Kosinski、D. Stillwell 和 T. Graepel。“从人类行为数字记录中可以预测私人特征和属性”。载于：《美国国家科学院院刊》110.15（2013），第5802–5805页。[72] A. Krizhevsky、I. Sutskever 和 G. E. Hinton。“使用深度卷积神经网络进行 ImageNet 分类”。载于：《神经信息处理系统进展》2012年，第1097–1105页。[73] B. Lakshminarayanan、A. Pritzel 和 C. Blundell。“使用深度集成进行简单且可扩展的预测不确定性估计”。载于：《神经信息处理系统进展》30（2017）。[74] S. Lapuschkin、A. Binder、G. Montavon、K.-R. Muller 和 W. Samek。“分析分类器：费舍尔向量和深度神经网络”。载于：IEEE计算机视觉与模式识别会议论文集。2016年，第2912–2920页。[75] A. Lavie 和 A. Agarwal。“METEOR：一种与人工判断高度相关的机器翻译自动评估指标”。载于：第二届统计机器翻译研讨会论文集。2007年，第228–231页。[76] J. Lee、M. Humt、J. Feng 和 R. Triebel。“以稀疏信息形式估计神经网络的模型不确定性”。载于：国际机器学习会议。PMLR，2020年，第5702–5713页。[77] S. Lee、S. Purushwalkam、M. Cogswell、D. Crandall 和 D. Batra。“为什么 M 个注意力头优于一个：训练多样化的深度网络集成”。2015年。arXiv：1511.06314。[78] M. Lewis。《语言模型解码·深度学习》。2020年4月20日。网址：https://atcold.github.io/pytorch-Deep-Learning/en/week12/12-2/（访问于2021年7月30日）。[79] J. Li、X. Chen、E. Hovy 和 D. Jurafsky。“可视化和理解自然语言处理中的神经模型”。2015年。arXiv：1506.01066。[80] C.-Y. Lin。“ROUGE：用于摘要自动评估的软件包”。载于：《文本摘要：拓展领域》。2004年，第74–81页。[81] T. Lin、Y. Wang、X. Liu 和 X. Qiu。“Transformer 综述”。2021年。arXiv：2106.04554。[82] H. Liu、Q. Yin 和 W. Y. Wang。“迈向可解释自然语言处理：用于文本分类的生成式解释框架”。2019年6月11日。arXiv：1811.00196。[83] J. Z. Liu、Z. Lin、S. Padhy、D. Tran、T. Bedrax-Weiss 和 B. Lakshminarayanan。“通过距离感知，使用确定性深度学习进行简单且有原则的不确定性估计”。2020年10月25日。arXiv：2006.10108。[84] S. M. Lundberg 和 S.-I. Lee。“解释模型预测的统一方法”。载于：第31届国际神经信息处理系统会议论文集。2017年，第4768–4777页。[85] A. Malinin 和 M. Gales。“先验网络的反向 KL 散度训练：改进不确定性和对抗鲁棒性”。2019年。arXiv：1905.13472。[86] P. H. Martins、Z. Marinho 和 A. F. Martins。“稀疏文本生成”。2020年。arXiv：2004.02644。[87] B. McCann、J. Bradbury、C. Xiong 和 R. Socher。“在翻译中学习：上下文化词向量”。载于：《神经信息处理系统进展》2017年，第6294–6305页。[88] P. McClure 和 N. Kriegeskorte。“通过深度神经网络中的采样稳健地表示不确定性”。2016年。arXiv：1611.01639。[89] L. McInnes、J. Healy 和 J. Melville。“UMAP：用于降维的统一流形逼近与投影”。2018年。arXiv：1802.03426。[90] C. Meister、T. Vieira 和 R. Cotterell。“如果束搜索是答案，那么问题是什么？”2021年1月17日。arXiv：2010.02650 [cs]。[91] P. Mertikopoulos、N. Hallak、A. Kavis 和 V. Cevher。“随机梯度下降在非凸问题中的几乎必然收敛性”。2020年6月19日。arXiv：2006.11144。[92] D. Metzler、Y. Tay、D. Bahri 和 M. Najork。“重新思考搜索：将专家培养成博学者”。2021年5月5日。arXiv：2105.02274 [cs]。[93] T. Mikolov、K. Chen、G. Corrado 和 J. Dean。“在向量空间中高效估计词表示”。2013年。arXiv：1301.3781。

[94] G. A. Miller。“WordNet：英语词汇数据库”。载于：《ACM通讯》38.11（1995），第39–41页。[95] C. Molnar。《可解释机器学习》。2022年1月21日。网址：https://christophm.github.io/ interpretable-ml-book/（访问于2022年1月26日）。[96] R. Moradi、R. Berangi 和 B. Minaei。“深度模型正则化策略综述”。载于：《人工智能评论》53.6（2020），第3947–3986页。[97] S. Morgan。Tensorflow/Addons。tensorflow，2020年12月1日。网址：https://github.com/tensorflow/addons/blob/0c0fd8dfb4427df6b824c88f700ba5c7efd43bec/tensorflow_addons/optimizers/lamb.py（访问于2021年11月8日）。[98] Z. Nado。《深度学习中不确定性与鲁棒性的基准》。Google AI 博客。2021年10月14日。网址：http://ai.googleblog.com/2021/10/baselines-for-uncertainty-and.html（访问于2021年10月25日）。[99] Z. Nado 等人。“不确定性基准：深度学习中不确定性与鲁棒性的基准测试”。2021年6月7日。arXiv：2106.04015。[100] R. Nakano 等人。“WebGPT：带有人工反馈的浏览器辅助问答”。2021年。arXiv：2112.09332。[101] S. Narang 等人。“Transformer 修改能否跨实现和应用迁移？”2021年9月10日。arXiv：2102.11972 [cs]。[102] R. M. Neal。《通过混合蒙特卡洛方法进行反向传播网络的贝叶斯训练》。技术报告 CRG-TR-92-1，计算机科学系，多伦多大学。Citeseer，1992年。[103] C. Nemeth 和 P. Fearnhead。“随机梯度马尔可夫链蒙特卡罗”。载于：《美国统计学会会刊》116.533（2021），第433–450页。[104] Z. Niu、G. Zhong 和 H. Yu。“深度学习注意力机制综述”。载于：《神经计算》452（2021），第48–62页。[105] K. Osawa、S. Swaroop、A. Jain、R. Eschenhagen、R. E. Turner、R. Yokota 和 M. E. Khan。“基于贝叶斯原理的实用深度学习”。2019年。arXiv：1906.02506。[106] Y. Ovadia 等人。“你能信任模型的不确定性吗？数据集偏移下的预测不确定性评估”。2019年。arXiv：1906.02530。[107] G. Paass。“通过自助法评估和改进神经网络预测”。载于：《神经信息处理系统进展》。Citeseer，1993年，第196–203页。[108] G. Paass 和 J. Kindermann。“应用于信用评分的具有重叠叶节点的贝叶斯分类树”。载于：《知识发现与数据挖掘研究与发展》，由 X. Wu、R. Kotagiri 和 K. B. Korb 编。计算机科学讲义。柏林、海德堡：施普林格，1998年，第234–245页。ISBN：978-3-540-69768-8。DOI：10.1007/3-540-64383-4_20。[109] Paperswithcode。《浏览人工智能领域的最先进成果》。2019年。网址：https://paperswithcode.com/sota。[110] K. Papineni、S. Roukos、T. Ward 和 W.-J. Zhu。“BLEU：一种机器翻译自动评估方法”。载于：《计算语言学协会第40届年会论文集》。2002年，第311–318页。[111] K. Pearson。“关于空间中点集系统的最佳拟合直线和平面”。载于：《伦敦、爱丁堡和都柏林哲学杂志与科学杂志》2.11（1901），第559–572页。[112] J. Pérez、J. Marinković 和 P. Barceló。“现代神经网络架构的图灵完备性”。2019年。arXiv：1901.03429。[113] C. Pierse。Transformers Interpret。0.5.2版。2021年2月。网址：https://github.com/cdpierse/ transformers-interpret（访问于2021年11月23日）。[114] Pytorch。PyTorch。2019年。网址：https://pytorch.org/。[115] M. Qudar 和 V. Mago。《语言模型综述》。2020年9月7日。网址：https://www.researchgate.net/publication/344158120_A_Survey_on_Language_Models/。[116] A. Radford、K. Narasimhan、T. Salimans 和 I. Sutskever。“通过生成式预训练改进语言理解”。载于：（2018年）。[117] A. Radford、J. Wu、D. Amodei、D. Amodei、J. Clark、M. Brundage 和 I. Sutskever。“更好的语言模型及其影响”。载于：《OpenAI博客》（2019年）。网址：https: //openai.%20com/blog/better-language-models。

[118] A. Radford、J. Wu、R. Child、D. Luan、D. Amodei 和 I. Sutskever。《语言模型是无监督多任务学习者》。载于：《OpenAI博客》1.8（2019），第9页。[119] S. Rajbhandari、J. Rasley、O. Ruwase 和 Y. He。《ZeRO：面向万亿参数模型训练的内存优化》。2020年5月13日。arXiv：1910.02054v3。[120] P. Rajpurkar、J. Zhang、K. Lopyrev 和 P. Liang。《SQuAD：用于文本机器阅读理解的100,000+ 个问题》。2016年。arXiv：1606.05250。[121] A. Ramesh、M. Pavlov、G. Goh 和 S. Gray。《 {DALL·E}：从文本创建图像》。2021 年 1 月 5 日。网址：https://openai.com/blog/dall-e/。[122] J. Rasley。DeepSpeed。微软，2021年12月20日。网址：https://github.com/microsoft/DeepSpeed（访问于2021年12月20日）。[123] M. T. Ribeiro、S. Singh 和 C. Guestrin。《机器学习的模型无关可解释性》。2016年。arXiv：1606.05386。[124] A. Rogers、O. Kovaleva 和 A. Rumshisky。《BERT学入门：我们对{BERT}工作原理的了解》。载于：《计算语言学协会会刊》8（2021），第842–866页。[125] S. Rönnqvist、J. Kanerva、T. Salakoski 和 F. Ginter。《多语言 BERT 擅长语言生成吗？》 2019 年。arXiv：1910.03806。[126] A. Rush。《带注释的Transformer》。载于：NLP 开源软件研讨会论文集（NLP-OSS）。澳大利亚墨尔本：计算语言学协会，2018年7月，第52–60页。doi：10.18653/v1/W18-2509。[127] A. B. Sai、A. K. Mohankumar 和 M. M. Khapra。《自然语言生成系统所用评估指标综述》。2020年。arXiv：2008.12009。[128] E. F. Sang 和 F. De Meulder。《CoNLL-2003 共享任务简介：语言无关的命名实体识别》。2003年。arXiv：cs/0306050。[129] S. Serrano 和 N. A. Smith。《注意力是否可解释？》2019年。arXiv：1906.03731。[130] D. So、Q. Le 和 C. Liang。《进化型Transformer》。载于：国际机器学习会议。PMLR，2019年，第5877–5886页。[131] L. Spinney。《我们是否正见证后理论科学的黎明？》载于：《卫报》。科技版（2022年1月9日）。ISSN：0261-3077。网址：
h
t
t
p
s://
w
w
w.theguardian.co
m/technology/2022/jan/09/are-we-witnessing-the-dawn-of-post-theory-science（访问于2022年1月11日）。[132] M. Sundararajan、A. Taly 和 Q. Yan。《深度网络的公理归因》。载于：国际机器学习会议。PMLR，2017年，第3319–3328页。[133] C. Szegedy、V. Vanhoucke、S. Ioffe、J. Shlens 和 Z. Wojna。《重新思考用于计算机视觉的 Inception 架构》。载于：IEEE 计算机视觉与模式识别会议论文集。2016年，第2818–2826页。[134] Y. Tay、D. Bahri、D. Metzler、D.-C. Juan、Z. Zhao 和 C. Zheng。《Synthesizer：重新思考Transformer模型中的自注意力》。2021年5月24日。arXiv：2005.00743 [cs]。[135] A. Taylor、M. Marcus 和 B. Santorini。《宾夕法尼亚树库：概述》。载于：《树库》（2003），第5–22页。[136] Tensorflow。《集成梯度》 |TensorFlow Core。TensorFlow。2021年11月25日。网址：https://www.tensorflow.org/tutorials/interpretability/integrated_gradients（访问于2021年12月6日）。[137] Tensorflow。Tensorflow网站。2019年。网址：https://www.tensorflow.org/。[138] tensorflow。《使用 SNGP 的不确定性感知深度学习》 | TensorFlow Core。TensorFlow。2021年。网址：https://www.tensorflow.org/tutorials/understanding/sngp（访问于2021年7月25日）。[139] E. Tjoa 和 C. Guan。《可解释人工智能（XAI）综述：迈向医疗可解释人工智能》。载于：《IEEE 神经网络与学习系统汇刊》（2020）。[140] L. van der Maaten 和 G. Hinton。《使用 t-SNE 可视化数据》。载于：《机器学习研究杂志》9（2008年11月），第2579–2605页。[141] A. Vaswani 等人。《Attention Is All You Need》。载于：神经信息处理系统进展。2017年，第5998–6008页。[142] J. Vig。《Transformer模型中注意力的多尺度可视化》。2019年。arXiv：1906.05714。[143] J. Vig。BertViz。2021年11月23日。网址：https://github.com/jessevig/bertviz（访问于2021年11月23日）。[144] J. Vig。BERTVIZ：用于可视化 BERT模型中多头自注意力的工具。2019年。网址：https://debug-ml-iclr2019.github.io/cameraready/DebugML-19_paper_2.pdf。

[145] Wang。SuperGLUE 基准。SuperGLUE 基准。2021年。网址：https://super.gluebenchmark.com/（访问于2021年2月23日）。[146] A. Wang、A. Singh、J. Michael、F. Hill、O. Levy 和 S. R. Bowman。《GLUE：自然语言理解的多任务基准与分析平台》。2019年2月22日。arXiv：1804.07461。[147] D. Wang、C. Gong、M. Li、Q. Liu 和 V. Chandra。《AlphaNet：使用 Alpha 散度改进超网络训练》。2021年。arXiv：2102.07954。[148] D. Wang、M. Li、C. Gong 和 V. Chandra。《AttentiveNAS：通过注意力采样改进神经架构搜索》。载于：IEEE/CVF 计算机视觉与模式识别会议论文集。2021年，第6418–6427页。[149] H. Wang、Z. Wu、Z. Liu、H. Cai、L. Zhu、C. Gan 和 S. Han。《HAT：面向高效自然语言处理的硬件感知 Transformer》。2020年。arXiv：2005.14187。[150] M. Welling 和 Y. W. Teh。《通过随机梯度朗之万动力学进行贝叶斯学习》。载于：第28届国际机器学习会议论文集（ICML-11）。2011年，第681–688页。[151] L. Weng。《注意力？注意力！》Lil'Log。2018年6月24日。网址：https://lilianweng.github.io/2018/06/24/attention-attention.html（访问于2021年11月19日）。[152] F. Wenzel 等人。《深度神经网络中的贝叶斯后验究竟有多好？》2020年。arXiv：2002.02405。[153] G. Wiedemann、S. Remus、A. Chawla 和 C. Biemann。《BERT讲得通吗？使用上下文化嵌入进行可解释的词义消歧》。2019年。arXiv：1909.10430。[154] Y. Wu 等人。《Google神经机器翻译系统：弥合人工翻译与机器翻译之间的差距》。2016年。arXiv：1609.08144。[155] F. Xu、H. Uszkoreit、Y. Du、W. Fan、D. Zhao 和 J. Zhu。《可解释人工智能：历史、研究领域、方法与挑战简述》。载于：CCF国际自然语言处理与中文计算会议。施普林格，2019年，第563–574页。[156] Y. Xu 等人。《GSPMD：面向机器学习计算图的通用可扩展并行化》。2021年12月23日。arXiv：2105.04663 [cs]。[157] Z. Yang、Z. Dai、R. Salakhutdinov 和 W. W. Cohen。《突破 Softmax 瓶颈：高秩 RNN 语言模型》。2017年。arXiv：1711.03953。[158] Z. Yang、Z. Dai、Y. Yang、J. Carbonell、R. R. Salakhutdinov 和 Q. V. Le。《XLNet：用于语言理解的广义自回归预训练》。载于：神经信息处理系统进展。2019年，第5753–5763页。[159] Y. You 等人。《深度学习的大批量优化：76分钟训练好BERT》。2019年。arXiv：1904.00962。[160] C. Yun、S. Bhojanapalli、A. S. Rawat、S. J. Reddi 和 S. Kumar。《Transformer是序列到序列函数的通用逼近器吗？》2019年。arXiv：1912.10077。[161] C. Yun、Y.-W. Chang、S. Bhojanapalli、A. S. Rawat、S. J. Reddi 和 S. Kumar。《$ O (n) $ 连接已足够强大：稀疏Transformer的通用逼近性》。2020年。arXiv：2006.04862。[162] B. Zhang 和 R. Sennrich。《均方根层归一化》。2019年。arXiv：1910.07467。[163] C. Zhang 等人。《ResNet 还是 DenseNet？为 ResNet 引入稠密捷径连接》。载于：IEEE/CVF 冬季计算机视觉应用会议论文集。2021年，第3550–3559页。[164] T. Zhang、V. Kishore、F. Wu、K. Q. Weinberger 和 Y. Artzi。《BERTScore：使用 BERT 评估文本生成》。2020年2月24日。arXiv：1904.09675。[165] W. Zhu、X. Wang、X. Qiu、Y. Ni 和 G. Xie。《AutoRC：通过架构搜索改进基于 BERT 的关系分类模型》。2020年。arXiv：2009.10680。[166] M.-A. Zöller 和 M. F. Huber。《自动化机器学习框架的基准测试与综述》。载于：《人工智能研究杂志》70（2021），第409–472页。

第3章
改进预训练语言模型

摘要 本章介绍了多种提升预训练语言模型（PLM）性能的方法，包括 BERT 的变体、类似 GPT 的自回归语言模型，以及类似 Transformer 的序列到序列模型。首先，我们可以修改预训练任务，以尽可能多地学习语言的句法和语义。然后，我们可以延长输入序列的长度，以便处理更长的输入。多语言模型会同时使用不同语言的文本进行训练。最重要的是将更多知识纳入预训练语言模型，从而生成更准确的预测。事实证明，通过增加参数数量、训练数据规模和计算开销，总能提升模型的性能。多种微调策略可用于使模型适应特定任务。此外，还可以通过少样本提示引导模型完成具体任务。这对于规模更大的预训练语言模型尤其有成效，因此这类模型被称为基础模型。

关键词： 预训练目标、输入大小、多语言模型、长距离依赖、额外知识、微调

本章介绍了多种提升预训练语言模型 （PLM）性能的方法，即 BERT 的变体、类似 GPT 的自回归语言模型，以及类似 Transformer 的序列到序列模型。当这些模型具有大量参数时，可以通过输入提示指导它们解决新任务，这类模型称为基础模型。

- **预训练任务的修改**。在使用大型语料库进行预训练时，PLM 应尽可能多地学习语言的句法和语义。通过调整和改进预训练目标，可以显著提升 PLM 的性能，如第 3.1 节所示。
- **增大输入大小**。输入序列的长度限制了 PLM 能够纳入考量的上下文。这对于

故事生成等应用尤为重要。简单地增加输入长度并不可行，因为参数数量会随之呈二次增长。第 3.2 节探讨了为远距离词元建立稀疏注意力模式的替代方案。

- **多语言训练** 会让同一个模型同时使用不同语言进行训练。通过设置恰当的预训练目标，模型可以用所有语言生成联合语义表示。对于训练数据较少的语言，这种方法尤其能带来更好的结果，见第 3.3 节。

- **添加额外知识**。通过加入训练数据未涵盖的额外信息，可以增强预训练语言模型。由于参数数量有限，预训练语言模型无法记住训练数据中的所有细节，因此这很重要。此外，严格的规则通常只表现为较弱的关联，需要进一步强化。通过整合外部知识库 （知识库）中的事实和规则，或额外的文本集合，预训练语言模型可以获取必要的信息并保持内容更新，如第 3.4 节所示。

- **改变模型规模**。理论结果表明，随着预训练语言模型变大，模型性能会提升（基础模型）。因此，普遍趋势是增大模型规模，例如采用混合专家架构。另一方面，也可能需要降低预训练语言模型的计算开销和内存占用。正如第 3.5 节所述，有多种方法可以在不过多牺牲性能的情况下实现这一点。

- **特定应用的微调**。这可以采用不同策略进行，例如，执行多个微调步骤或多个微调任务。通常，可以通过提示指示较大的预训练语言模型执行特定任务，这类模型称为基础模型。此外，还可以优化少样本提示，以获得更恰当的模型响应。第 3.6 节对此进行了介绍。

请注意，对于大多数模型类型，几乎所有方案都可以组合使用，因此产生了数量庞大的模型变体，目前对此已有广泛讨论。

3.1 修改预训练目标

基础 BERT模型 [49] 有两个预训练任务：使用掩码语言模型（MLM）预测掩码词元，以及下一句预测（NSP）（第 2.1 节）。这些任务是通过启发式方法选定的，而损失函数和架构还有许多合理的选择。研究人员已经考察了许多替代性的训练目标、模型结构和注意力机制。本节将讨论 BERT 和 Transformer 架构的这些变体中最有前景的方案，并比较它们各自的优劣。

一个重要问题是输入序列的聚合层级。这里通常采用子词标记。一种选择是使用原始字母作为输入。然而，这可能会导致较高的计算负担，因为自注意力的计算成本会随着输入大小呈二次增长。另一种选择是利用领域适配知识，通过学习得到的分词方式或

表格 3.1 预训练语言模型的损失函数。序列表示为 $\boldsymbol{x} = (x_1, \ldots, x_T)$，而 $\boldsymbol{z} = (z_1, \ldots, z_R)$ 是一个相关序列，例如翻译。

Name	损失函数	描述
多变量分类	$L_{MC} = -$ 日志 $p(y\|\boldsymbol{x})$	对于每个训练样本（$\boldsymbol{x}$，y），例如
分类		逻辑分类器，第 1.3 节
NM 邻域	$L_{NM} = -\sum_{t=1}^{T}\sum_{i\in N(t)}$ 日志 $p(x_i\|x_t)$	对于邻域 $N(t)$ =
模型		$\{t-k, \ldots, t-1, t+1, \ldots, t+k\}$，例如 Word2Vec ，第1.5节
语言模型语言	$L_{LM} = -\sum_{t=1}^{T}$ 对数 $p(x_t\|\boldsymbol{x}_{<t})$	例如：循环神经网络 第1.6节，GPT 第2.2.2节
模型		
序列到序列 序列到	$L_{S2S} = -\sum_{t=1}^{n_z}$ 对数 $p(z_t\|\boldsymbol{z}_{<t}, \boldsymbol{x})$	对于输入序列 $\boldsymbol{x} = (x_1, \ldots\ldots, x_T)$ 和
序列		翻译 $\boldsymbol{z} = (z_1, \ldots\ldots, z_R)$ 第1.6节，2.3
模型		
掩码语言模型掩码	$L_{MLM} = -\sum_{t\in m(\boldsymbol{x})}$ 日志 $p(x_t\|\tilde{\boldsymbol{x}})$	$m(\boldsymbol{x})$ 包含掩码位置的索引
语言模型		$\boldsymbol{x}$中的词元。在 $\tilde{\boldsymbol{x}}$ 中，掩码词元 被替换为*MASK*，例如 BERT，见第 2.1 节
TLM 翻译	$L_{TLM} = -\sum_{t\in m(x)}\log\ p(x_t\|\tilde{\boldsymbol{x}})$	$m(\boldsymbol{x})$包含被掩码的索引
掩码语言		词元。$\tilde{\boldsymbol{x}}$ 包含一个句子及其
模型		翻译。掩码词元会被替换
		为 掩码，例如 mBERT，第 3.3 节
SBO 跨度	$L_{SMLM} = -\sum_{(i:\ j)\in m(\boldsymbol{x})}$ 日志 $p(\boldsymbol{x}_{i:j}\|\tilde{\boldsymbol{x}})$	$m(\boldsymbol{x})$ 包含以下跨度（i： j）
边界		$\boldsymbol{x}$ 中的掩码词元。在 $\tilde{\boldsymbol{x}}$ 中，掩码
目标		词元会被其他词元替换，例如
		SpanBERT，第 3.1.1 节
预训练语言模型排列	$L_{PLM} = -\sum_{t=1}^{T}$ 日志 $p(z_t\|\boldsymbol{z}_{<t})$	$\boldsymbol{z} = perm(\boldsymbol{x})$是 $\boldsymbol{x}$的一个排列，例如
语言模型		XLNet，第 3.1.1 节
NSP 下一句	$L_{NSP} = -$ 对数 $p(\xi\|\boldsymbol{x}, \boldsymbol{z})$	ξ=1 如果 $\boldsymbol{z}$ 之后的文本 x (else $\boldsymbol{z}$ 是随机的
预测		选择），例如 BERT，第 2.1 节
SOP 句子	$L_{SOP} = -$ 对数 $p(\xi\|\boldsymbol{x}, \boldsymbol{z})$	ξ=1 如果文本 $\boldsymbol{z}$ 之后 $\boldsymbol{x}$ （否则 $\boldsymbol{x}$ 之后 $\boldsymbol{z}$），例如
顺序预测		ALBERT，第 3.1.1 节
RTD 替换词元	$L_{RTD} = -$ 在 $\boldsymbol{x}$ 中随机选取的元素上取对数 $\sum_{t=1}^{T} p(x_t{=}\tilde{x}_t\|\tilde{\boldsymbol{x}})$	
词元检测		被替换，例如 ELECTRA，第 3.1.1 节

图像块嵌入（例如用于图像表示，见第 7.2 节）对输入序列进行建模。这些方法降低了输入复杂度，但可能会忽略输入中的有用信息 [19]。

3.1.1 类似 BERT 的自编码器

为了提升 BERT 的性能，人们提出了多种从未标注数据中获取知识的替代方案:

- RoBERTa 会在训练过程中动态更改掩码。
- ALBERT用矩阵乘积替代自注意力中的矩阵，并在所有层之间共享参数。

表 3.2 与 BERT 类似的自编码器。预训练和微调损失函数定义见表 3.1。基准测试数据仅供参考，因为它们取决于参数数量和计算开销。

模型	章节	预训练	微调	额外	基准测试
ELMo [156]	1.6	BiLM	MC	使用双向长短期记忆网络	GLUE 71.0
BERT [49]	2.1	掩码语言模型 + 下一句预测	多项选择	预测掩码词元	GLUE 80.5
RoBERTa [127]	3.1.1	MLM	MC	训练更长时间，使用新的掩码新训练轮次	GLUE 88.5
SpanBERT [98]	3.1.1	预训练语言模型、跨度边界目标	MC	预测词元跨度	GLUE 82.8
ELECTRA [223]	3.1.1	RTD	MC	替换词元检测	GLUE 89.4
StructBERT [39]	3.1.1	RTD	MC	重排打乱的词元	GLUE 89.0
ALBERT [113]	3.1.1	MLM + SOP	MC	因式分解嵌入，参数共享	GLUE 89.4
XLNET [240]	3.1.1	PLM	MC	预测置换后的词元	GLUE 90.5
DeBERTa [76]	3.1.1	MLM	多项选择，序列到序列	解耦注意力	GLUE 90.0
产品密钥 [112]	3.1.1	MLM	MC	最近邻	-
UniLM [8]	3.1.3	MLM、LM	MC、LM	单向和双向	GLUE 87.3
BigBird [247]	3.2.1	MLM	多项选择、序列到序列	稀疏注意力机制	TriviaQA 84.5

- 对单个掩码词元进行预测可以得到推广。SpanBERT对词元跨度进行掩码并预测它们。ELECTRA检测任意位置上被随机替换的词元。XLNet对句子中词元的顺序进行置换，并像语言模型一样从左到右预测词元。
- DeBERTa 将内容嵌入与位置嵌入解耦。

详细信息见以下段落。表 3.2 列出了具有代表性的自编码器。相应的损失函数定义于（表 3.1）中。可以根据它们在自然语言理解任务（第 2.1.5 节）上的性能进行比较，例如 GLUE [218]。

RoBERTa [127]是通过调整预训练过程的部分环节而增强的 BERT模型。作者通过以下更改改进了 $\text{BERT}_{\text{BASE}}$ 架构：(1) 不再在所有训练轮次中使用相同的掩码，而是使用不同的掩码复制训练序列。(2) 移除下一句预测目标，并发现当一个批次中的所有句子都来自同一文档时，性能最佳。(3) 更大的批次和更大的步长会提升掩码语言模型任务的困惑度以及下游任务性能。(4) 将训练数据增加十倍至 160GB，并使用大批次进行训练。最终模型在 *GLUE*（语言理解[217]）以及阅读理解任务 *RACE* 和*SQuAD* [173]上取得了令人瞩目的最先进水平结果 88.5。

SpanBERT [98] 提出了一种跨度级预训练方法。预训练时，不再对单个词元进行掩码，而是对一个或多个完整单词组成的跨度进行掩码，覆盖约 15% 的词元。文中还提出了一种新的跨度边界目标（SBO）：预测掩码跨度内部的词元时，仅使用跨度边界外紧邻词元的表示，并结合

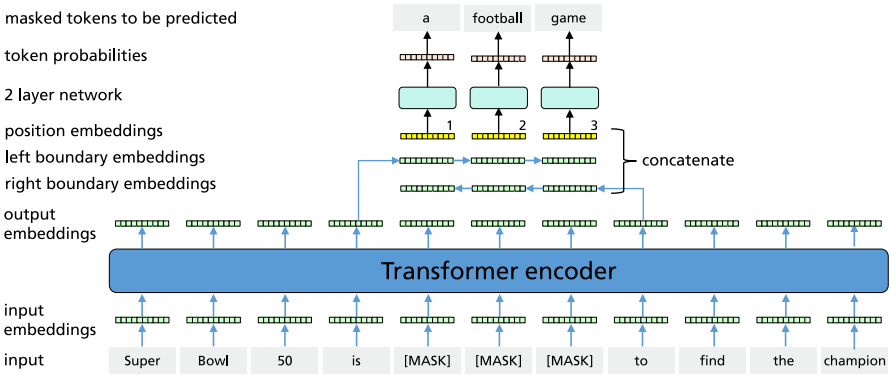


图 3.1 SpanBERT [98] 将跨度边界之外的嵌入与位置嵌入进行拼接。利用这一输入，一个 2 层模型可以预测掩码词元的概率。

位置信息。详细信息见图 3.1。SBO 与常用的掩码语言模型目标结合使用。最后，作者像 [127] 中一样省略了下一句预测任务，仅使用单个文本片段/句子进行训练。作者发现，随机跨度掩码比对语言单位进行掩码更有效。SpanBERT 的配置与 BERT $_{\text{LARGE}}$ 相同，并在 BooksCorpus 和英文维基百科上进行预训练。SpanBERT 在 *OntoNotes* 共指任务 [164], 中取得了新的最先进水平，F1值达到 79.6%。该任务要求识别代词及其对应的名词，或指代同一对象的两个短语（第 5.4.1 节）。

StructBERT [223] 通过增加预测打乱的词元三元组顺序的任务，增强了原始 BERT 掩码语言模型目标。此外，还必须检测三个句子的顺序。在使用参数数量相同的模型时，与 BERT 和 RoBERTa 相比，StructBERT 可将 GLUE 上的最先进水平分别提升至 83.9 和 89.0。

Electra [39] 提出了一种名为替换词元检测（RTD）的新预训练任务。在该论文中，将一个使用掩码语言模型损失进行训练的生成器网络与一个判别器网络相结合。输入序列中的一些词元会被一个小型语言模型生成的合理替代项所替换（该模型的规模约为判别器的 1/4）。判别器网络必须预测每个词元是否为替代项。这一破坏过程解决了 BERT 中的不匹配问题：在预训练期间会出现*MASK* 词元，但在微调期间不会出现。该模型利用所有输入词元进行学习，而不只是利用少量的掩码子集，因此在计算上比 BERT 和 RoBERTa 等模型更高效，同时在多个任务上表现更好，例如在 GLUE语言理解任务上达到 89.4%。

ALBERT（轻量版 BERT） [113] 采用两种参数缩减技术，以解决 BERT 内存消耗巨大、训练速度缓慢的问题。第一项改进是将 WordPiece 嵌入的维度与 BERT 隐藏层的维度解耦

。作者将嵌入矩阵 M进行因式分解$M = A * B$，使得 A 和 B 中的参数总数远少于 M中的参数数量。第二项改进是在 BERT 的所有层之间共享全部参数。研究表明，这有助于稳定训练；即使增加层数，参数数量也能保持不变。除了这两项改进，ALBERT 还引入了一种新的句子顺序预测（SOP）任务。具体来说，模型需要预测两个句子的顺序是正确还是颠倒。作者报告称，与 BERT 的下一句预测（NSP）任务相比，该任务提高了准确率；NSP 任务可以通过比较两个句子的主题来完成。不过，这是否是将文本结构融入训练的最佳方式，目前仍不清楚。ALBERT 在 GLUE 和 SQuAD 上取得了新的最先进水平。

XLNet执行自回归预训练任务，而不是预测掩码词 [240]。这解决了 BERT 的 $[MASK]$ 词元仅在预训练期间出现、而不会在微调期间出现的问题。序列中的词语，例如“The_1老鼠$_2$ 喜欢$_3$ 奶酪$_4$”，会通过随机排列与其位置信息（索引）一同重新排序，例如“奶酪$_4$ The_1 喜欢$_3$ 老鼠$_2$”。该任务是依次预测排列后序列中的词元，类似于 GPT 语言模型。模型需要进行如下预测，例如 p(老鼠|2，奶酪$_4$, The_1, 喜欢$_3$)。需要注意的是，模型还必须知道待预测词语的位置，此处为 2。然而，Transformer 通过求和将位置信息与内容信息混合在一起。因此，位置信息无法与词元嵌入分离。

因此，作者决定计算一种额外的自注意力嵌入，称为查询流。查询流作为查询时只接收目标位置，然后可以利用键向量和值向量计算注意力（第 2.1.1 节）。所得嵌入编码了待预测词元的位置以及与其他词元的相关性，但不包含该词元的内容信息。该信息可以作为输入添加到模型中。普通自注意力和查询流使用相同的参数矩阵 Q （查询）、K （键）、 V （值）。为了节省训练开销，XLNet 只预测排列后序列末尾的少数几个词元。此外，XLNet 将 Transformer-XL 的分段循环机制和相对编码方案（第 3.2.2 节）整合到预训练中，根据实证结果，这尤其提升了涉及较长文本序列的任务的性能。

预测一个词元时，可以使用其前后词元的信息。因此，该模型是一个双向编码器。对于 BERT，如果两个词元“New”和“$York$”被掩码，则两个词语会被独立预测，从而忽略有价值的信息。相比之下，XLNet 能够正确处理掩码词元之间的依赖关系。XLNet 在许多任务上的表现都优于 BERT 和 RoBERTa，例如 GLUE语言理解任务、SQuAD 等阅读理解任务（第 2.1.5 节），以及$IMDB$ 等文本分类任务（电影评论分类）[130]。

乘积键 [112] 用最近邻搜索替代点积注意力。查询 $\boldsymbol{q}_r$ 被拆分为两个子查询 $\boldsymbol{q}_r^{[1]}$和 $\boldsymbol{q}_r^{[2]}$。针对每个子查询，选出最接近的子键 $\boldsymbol{k}_i^{[1]}$和 $\boldsymbol{k}_j^{[2]}$。根据子键的 k^2种组合，可以高效计算出最大的点积，并选出其中最高的 k

组合。结果经过 softmax 函数归一化，并用于计算值向量的加权和。在优化过程中，只有 k 最优键会受到影响，从而降低训练开销。该方法只需极少的计算开销，就能构建规模非常大的 Transformer。作者使用 12 层模型，便达到了与 24 层 BERT模型相同的性能，而计算时间仅为其一半。在对 Transformer架构 进行全面比较时 [142]，该方法使 SuperGLUE 自然语言理解任务（第 4.1.2 节）的得分从标准 T5模型的 71.7% 提高到 75.2%。

DeBERTa [76] 采用了解耦注意力 机制，其中每个单词由两种不同类型的向量表示，分别编码内容和位置。词元之间的注意力权重使用不同的矩阵计算，分别对应内容和相对位置。此外，DeBERTa 在最后一层加入绝对词位置，以捕捉句子中的不同句法角色。在微调过程中，该模型采用“对抗式”训练方法，将嵌入归一化为概率向量。随后，模型经过训练，能够抵抗嵌入所受到的小幅扰动。根据作者的说法，这提高了微调模型的性能。该模型的1.5B参数大型版本在多个应用领域中表现优异，例如自然语言理解（第4.1.2节）。在*SuperGLUE* 基准 [219] 上，DeBERTa 首次超越了人类表现，将宏平均分数提高至89.9%。

Bengio 等人 [12] 认为，表示（例如嵌入）应当是解耦的，并且应在嵌入向量的不同部分表示不同的内容方面，例如句法、风格和语义。Locatello 等人 [129]已证明，这无法通过无监督方式实现。因此，必须使用某种明确的监督或先验信息，才能生成嵌入的可解释子向量。

DeBERTaV3 [75]用 ELECTRA 的替换词元检测（RTD）（第 3.1.1 节）替代了 DeBERTa 的掩码语言建模损失。此外，还采用了一种新的梯度解耦嵌入共享方法，同时提高了训练效率和预训练模型的质量。其最大版本拥有 128k 词元的词汇表、24 层和 3.04 亿个参数。在经过微调的 GLUE 基准测试中，该模型将得分提高了 1.4%，达到新的最先进水平 91.4%。该模型的多语言版本 $\text{mDeBERTa}_{\text{BASE}}$ 在 $\text{XNLI}_{\text{BASE}}$ 任务（第 3.3.1 节）上的跨语言迁移准确率方面超越了 $\text{XLM-R}_{\text{BASE}}$ 3.6%。

3.1.2 类似 GPT 的自回归语言模型

通过增加参数数量和训练集规模，GPT模型的能力可以得到显著提升。表3.3给出了概览。

GPT-3 [25] 是一种维度极大的语言模型。其最大版本有 96 层、96 个注意力头、1750 亿个参数，能够处理长度为 2,048 的序列。它使用约 5000 亿个词元的文本集合进行训练，这些文本来自书籍、维基百科和网页。其架构的详细信息尚未公开。GPT-3

表3.3 类似于 GPT 的自回归语言模型（LM）。“详细信息”提供参数数量和具体特征。“基准测试”数据仅供参考，因为它们取决于所选参数数量和计算量。

模型	Sec.	详细信息	基准测试
GPT-2 [167]	2.2	用于生成文本的 1.6B 语言模型	Lambada 零样本 63.2%
Retro [21]	6.2.3	结合检索生成文本的 7B 语言模型	Lambada 73.0%
Megatron-LM [193]	3.1.2	用于生成文本的83亿参数语言模型	Lambada 66.5%
Turing-NLG [179]	3.1.2	用于生成文本的170亿参数语言模型	Lambada 68.0%
Chinchilla [83]	3.1.2	700亿参数语言模型生成文本	Lambada 零样本 77.4%
GPT-3 [25]	3.1.2	175B 长序列语言模型生成文本	Lambada 零样本 76.2%
WebGPT [25]	6.2.3	175B GPT-3 + 必应搜索引擎	与 GPT-3 相同
InstructGPT [151]	3.6.5	175B 针对指令微调的 GPT-3，与 GPT-3 相同	
OPT [151]	3.1.2	类似于 GPT-3 的免费 175B 语言模型	Lambada 零样本 74.7%
BLOOM [151]	3.1.2	面向欧洲语言的1760亿参数语言模型	Lambada 零样本 67.2%
PanGu-α [248]	3.1.2	2000亿参数长序列语言模型生成文本	中文基准测试
Gopher [168]	3.1.2	2800亿参数语言模型生成文本	Lambada 零样本 74.5%
MT-NLG [4]	3.1.2	5300亿参数Megatron变体	Lambada 76.6%
PaLM [35]	3.1.2	540B 共享键值投影	Lambada 零样本 **77.9%**
GLaM [51]	3.5.2	1,200B 混合专家语言模型	Lambada 零样本 73.7%
悟道-2.0 [178]	3.5.2	1,750B 混合专家语言模型	Lambada：优于 Turing-NLG

在结构上与 GPT-2 相似，因此，其更高的准确率被归因于容量的提升和参数数量的增加。该模型在语言建模、问答等方面取得了前所未有的性能。表 3.4 汇总了部分结果，论文中还提供了更多结果 [25]。

GPT-3 能够生成流畅的文本，并涵盖海量的世界知识，如图 3.2 所示的示例。许多地方都可以找到生成文本的示例 [23, 149]。第4章讨论了预训练语言模型所捕获的知识数量和质量。与其他语言模型不同，只需提供几个句子作为指令，GPT-3 就能执行相当广泛的任务（少样本学习）。这种方式非常简单，可用于让 GPT-3 完成一些相当具体的任务，例如翻译成另一种语言、总结文档、纠正语法、围绕给定主题撰写论文等。详细信息见第 3.6.3 节。

2021年底，OpenAI 提供了一个 API，可使用用户特定的数据 [123]对 GPT-3 进行微调。这样，模型便能适应特定领域的语言，还可以针对特定分类任务进行准备。总体而言，这种方式比提示设计能获得更高质量的结果。此外，也不再需要少样本示例。第 3.6.2 节将讨论 GPT-3 微调的详细信息。表格 3.4 将 GPT-3 与其他较新的语言模型在若干热门基准测试上的表现进行了比较。新的 PaLM 模型优势明显。

GPT-J-6B是一种开源GPT模型，具有28层、16个注意力头、2,048的上下文长度和60亿个参数 [221]。它的性能与拥有67亿个参数的GPT-3版本相近。这里有一个交互式 Web 演示，用户可以在其中输入

输入： 标题：联合卫理公会教徒同意历史性分裂 副标题：反对同性婚姻的人将成立自己的教派 文章：

GPT-3 输出： Af

两经过两天的激烈辩论，联合卫理公会同意进行历史性分裂。预计这将促成一个新教派的创立，据《华盛顿邮报》报道，该教派将“在神学和社交方面持保守立场”。今年5月，参加该教会年度总议会的多数代表投票决定加强对任命 LGBTQ 神职人员的禁令，并制定新规则，“惩戒”主持同性婚礼的神职人员。但反对这些措施的人已经制定了新计划：他们表示将在2020年组建一个独立教派，并将其教会称为基督教卫理公会教派。

《华盛顿邮报》指出，这个拥有1250万名成员的教派在20世纪初曾是“美国最大的 新教教派”，但近几十年来一直在萎缩。这次新的分裂将是该教会历史上的第二次分裂。第一次发生在1968年，当时该教派约有10%的成员离开，组建了福音派联合弟兄会。《华盛顿邮报》指出，拟议中的分裂“发生在教会的关键时期，因为教会多年来一直在失去成员”，并且教会因 LGBTQ 人群在其中所扮演的角色而“被推向分裂的边缘”。同性婚姻并不是导致该教会分裂的唯一问题。2016年，该教派因跨性别神职人员的任命问题而分裂：北太平洋地区议会投票决定禁止他们担任神职人员，而南太平洋地区议会则投票决定允许他们担任神职人员。

图 3.2 GPT-3 根据输入生成的文本。经作者慷慨许可引用 [25,第 28]页。

根据提示生成续写文本 [220]。**GPT-Neo** [16] 是 GPT 的另一个免费版本，拥有 27 亿个参数。它使用 *Pile* 进行训练；Pile 是一个 825 GB 的数据集，包含来自 22 个不同来源的数据，包括学术来源（例如 arXiv）、互联网网页（例如 Stack Exchange）、字幕中的对话、GitHub 等。在某些自然语言理解任务中，它的表现优于参数规模相同的 GPT-3 版本 [89]。最近，**GPT-NeoX-20B** [215] 正式发布。它有 44 层，内部向量维度为 6,144，包含 64 个注意力头，训练时使用大小为 310 万的批次。在 LAMBADA 基准测试（第 4.1.3 节）中，任务是预测每个段落最后一句中缺失的最后一个单词，其准确率达到 72.0%。这一数值接近 GPT-3 的 75.2%。

Megatron-LM [193] 通过引入层内模型并行，高效扩展 GPT-2 和 BERT 等语言模型。作者将自注意力头和前馈层分配到不同的 GPU 上，从而减轻单个 GPU 的内存负担。他们提出了一种拥有 8.3B 个参数的 GPT 变体，以及一个类似 BERT、拥有 3.9B 个

参数的模型。该方法的亮点包括使用 512 个 GPU 时达到 76% 的扩展效率。他们的 GPT 模型将 WikiText 上的困惑度从 15.8 降至 10.8，达到最先进水平；他们的 BERT 模型则将 RACE（阅读理解）的准确率提升至 90.9%。

Jurassic-1 [122] 是一种自回归语言模型，与 GPT-3 类似，拥有 1780 亿个参数。作者选择了 256K 的词元词汇表，而非 GPT-3 使用的 50K 词元词汇表，其中还包括命名实体和常见短语等高频多词表达式。与 GPT-3 相比，该模型能用少 28% 的词元表示训练文本。因此，采用相同架构时，该模型处理查询的速度可提高至 1.4× 倍。该模型使用的最大序列长度为 2,048 个词元。尽管词汇表更大，输入嵌入仅占全部参数的 2%。该模型使用从公开文本语料库中提取的 3000 亿个词元进行训练，最终批量大小为 320 万个词元。

PanGu-α [248] 是华为研发的一种模型，与 GPT-3 类似，参数量最高可达 2000 亿。该模型使用 1.1 TB 中文文本进行训练，并在零样本、单样本和少样本设置下应用于大量任务，无需进行任何微调。该模型的性能与 GPT-3 相当。

OPT-175B（开放式预训练变换器）[253] 是 Meta 开发的一套 GPT 模型，共有 8 个，参数数量从 1.25 亿到 1750亿参数不等。这些模型使用包含 1800 亿词元的公开数据集进行训练。最大的模型有 96 层，每层各有 96 个注意力头。尽管 OPT-175B 的参数数量与 GPT-3 相同，但其训练所需的计算开销仅为 GPT-3 的七分之一。该模型在 16 项自然语言处理任务上进行了评估，性能与 GPT-3 大致相同（表 3.4 ）。所有参数数量不超过 300 亿的训练模型均可免费获取。为避免用于生产假新闻，1750亿参数的大型模型仅应学术研究人员申请提供。该模型只需使用 16 个 NVIDIA V100 图形处理器即可进行训练和部署。表 3.4 列出了一些基准测试结果。

BLOOM [139] 是一种具有 1760 亿参数的自回归大型语言模型。它有 70 层，每层有 112 个注意力头，词元序列长度为 2,048。该模型由 BigScience 倡议组织的 1,000 多名人工智能研究人员开发，旨在为所有想要尝试的人提供一个免费的大型语言模型。其训练数据涵盖 46 种自然语言（英语 30%、中文 16%、法语 12%、西班牙语 11%、……）以及 11% 的代码（Java、PHP、……），共计 3500 亿词元。这个拥有 1760 亿参数的 BLOOM 模型使用 Megatron-DeepSpeed 库进行训练 [26] ，该库提供不同类型的并行方式。该模型可在 8 个大型图形处理器上进行评估。因此，BLOOM 是可用于研究用途的最大训练模型之一。表 3.4 列出了一些基准测试结果。

Gopher [168] 采用了经过两处修改的 GPT-2 架构。为实现正则化，作者使用 RMSNorm（第 2.4.2 节）替代了 LayerNorm，并采用相对位置编码方案 [44]替代绝对位置编码。Gopher 拥有 80 层、128 个注意力头和 2800 亿个参数。所有模型均使用 3000 亿个词元进行训练，上下文窗口为 2,048 个词元，批量大小最高可达 600 万个词元。对于大型模型，使用了 16 位浮点数，以减少内存占用并提高训练吞吐量。

表格 3.4 比较 PaLM、GPT-3、Chinchilla、Gopher、OPT、GLaM 和 BLOOM 的不同版本在多项热门基准测试上的表现，这些基准测试涵盖文本补全、代词共指、常识推理和问答（QA） [22, 25, 35, 51]。FLOPS 衡量以每秒浮点运算次数计的计算开销。

	PaLM	PaLM	PaLM	GPT-3	Chinchilla	Gopher	OPT	GLaM	BLOOM
模型规模（十亿参数）	8	62	540	175	70	280	175	1200	176
训练词元数量（十亿）	780	795	780	400	1400	300	180	1600	350
训练开销（10^{21} FLOPS）	37.4	295.7	2527	314.0	588.0	504.0	≈ 50	≈ 105	
Lambada 零样本（文本补全）	69.5	75.4	**77.9**	76.2	77.4	74.5		73.7	67.2
HellaSWAG 零样本（文本补全）	68.7	79.7	**83.4**	78.9	80.8	79.2	79.0	77.1	73.0
PIQA 零样本（常识）	77.1	80.5	**82.3**	80.5	81.8	81.8	78.5	80.4	
Winogrande 零样本（共指）	66.3	77.0	**81.1**	70.2	74.9	70.1	74.0	73.4	70.1
BoolQ 零样本（问答）	68.3	84.8	**88.0**	60.5	83.7	79.3	64.0	83.0	
Natural Questions 零样本（问答）	8.4	18.1	**21.2**	14.6	16.6	10.1		21.5	
NaturalQuestions少样本（问答）	14.6	27.6	**36.0**	29.9	31.5	24.5			
Trivia QA 零样本（问答）	39.5	67.3	**76.9**	64.3	67.0	52.8		68.0	
Trivia QA 少样本（问答）	48.5	72.7	**81.4**	71.2	73.2	63.6			
平均任务指标	51.2	64.8	**69.8**	60.7	65.2	59.5			

为了评估模型规模的影响，作者训练了参数数量不同的六个模型版本。作者在表 4.3 中所述的 152 项任务上进行了全面评估。Gopher 在 124 项任务中的 100 项上取得了改进。其中一项是*LAMBADA* 基准测试 [154]，Gopher 在该测试中以零样本方式获得了 74.5 分，仅略低于参数量为 5300 亿的*MT-NLG* 模型的 76.6 分 [106]。例如，在涵盖计量经济学和心理学等领域的人文学科中，Gopher 在全部 12 项基准测试中均达到最先进水平，并在其中 11 项基准测试中超过了最佳有监督结果。表 3.4 列出了一些结果，第 4.1.4 节则介绍了更多详细信息。

Chinchilla [83] 是一种中等规模的编码器模型，拥有 70B 个参数，其计算预算与规模更大的 Gopher 模型相同，但使用的数据量是后者的四倍。Chinchilla 的性能始终优于 Gopher（表格 3.4），并且在大量下游评测任务上的表现显著超过 GPT-3（175B）、Jurassic-1（178B）和 Megatron-Turing NLG（530B）。模型规模每翻一倍，训练词元的数量也应翻一倍。这一缩放率远高于 Kaplan 等人在第 3.5.1 节中预测的缩放率。 [102]

Turing-NLG [179] 推出了一种包含 78 个 Transformer 层、隐藏向量维度为 4256、拥有 28 个注意力头和 170 亿个参数的自回归语言模型。由于参数量超过 13 亿的模型无法装入单个拥有 32GB 内存的 GPU，因此必须在多个 GPU 之间进行并行化，或拆分成多个部分。Turing-NLG 利用了具有高通信带宽的最先进深度学习硬件、Megatron-LM 框架以及 DeepSpeed 库，后者进一步优化了训练速度并减少了所需资源。该模型在语言建模任务上取得了最先进水平的性能，并证明其在零样本问答和抽象式摘要方面同样有效。

它的后继模型 **MT-NLG** [4] 是一个拥有 105 层、5300 亿参数的编码器模型，训练时使用了 280 个图形处理器，批量大小高达 1920。与 GPT-3 类似，它提升了零样本、单样本和少样本任务的性能。对于 *LAMBADA*

基准测试 [154], 例如，模型需要预测段落中的最后一个单词（第 4.1.3 节）。在这项基准测试中，MT-NLG 将 GPT-3 的少样本准确率（86.4%）提升至最先进水平 87.2%。

PaLM [35] 是由 Google 开发的、拥有 540B 个参数的自回归语言模型。它有 118 层、48 个注意力头，输入序列长度为 2,048。此外，还有参数量为 8B 和 62B 的较小版本。它采用带有 SwiGLU 激活函数的标准自回归解码器，并为一层中的注意力头使用共享查询值投影，从而提高自回归解码速度。该模型在包含 780B 个词元的高质量数据集上进行训练，其中已过滤掉粗俗和有害语言。每个训练样本只使用一次。训练集包含社交媒体对话（50%）、多语言网页（27%）、书籍（13%）、源代码文件（5%）、多语言维基百科文章（4%）和新闻文章（1%）。训练使用了 3072 个张量处理单元，持续 1368 小时，最终产生的总排放量比乘坐飞机在旧金山和纽约之间进行一次直接往返航班的排放量高 50% [35, p。 18]。

PaLM 在数百项自然语言推理、数学、推理和知识密集型任务上进行了评估，并在绝大多数基准测试中达到了最先进的准确率，例如在 29 项评估最广泛的英语语言理解基准测试中，有 28 项取得了这一成绩（参见表 3.4）。这表明，对于大型基础模型而言，规模效应仍然成立。图 3.3 展示了 BIG-bench 数据上的结果，并与此前的模型进行了比较。PaLM 540B 5-shot 在 58 项共同任务中的 44 项上超过了此前的最先进水平，平均表现也显著优于其他模型（Gopher、Chinchilla、GPT-3）。此外，PaLM 540B 5-shot 的得分高于受邀解决相同任务的人类的平均得分。在 SuperGLUE 上进行微调后，该模型超越了最佳的仅解码器模型，并且能够与编码器-解码器模型相抗衡；总体而言，后者在微调方面表现更好。大量任务显示出随着模型规模扩大而产生的不连续改进，这意味着从较小版本到最大模型的性能提升高于预期。

PaLM 已在程序代码文档上进行微调。由此得到的模型称为 *PaLM-Coder* [35, p.23]。代码质量通过 pass@k指标进行衡量：对于测试集中的每个问题，PaLM-Coder 会生成 k 个源代码样本；如果任一样本能够解决该问题，则将其计为已解决。PaLM-Coder 能够完成多项基准任务，其 pass@1 值约为 50。文中还对 PaLM-Coder 模型的性质进行了详尽评估。

对于约四分之一的任务，作者观察到，当模型规模从 58B 增加到 540B 个参数时，准确率会出现不连续跃变，远远超出 Kaplan 等人提出的“幂律” [102] （第 3.5.1 节）。示例包括图 3.3 所示的“英语谚语”和“逻辑序列”。这表明，大型语言模型的新能力可能会在模型达到足够规模时涌现，并且这些能力还会在迄今研究过的模型规模之外继续发展。

训练数据中有 22% 是多语言文档。在不同语言之间进行翻译时，少样本 PaLM 模型的表现接近甚至超过微调后的最先进水平。对于英法语翻译，PaLM 540B 少样本模型的 BLEU 分数达到 44.0

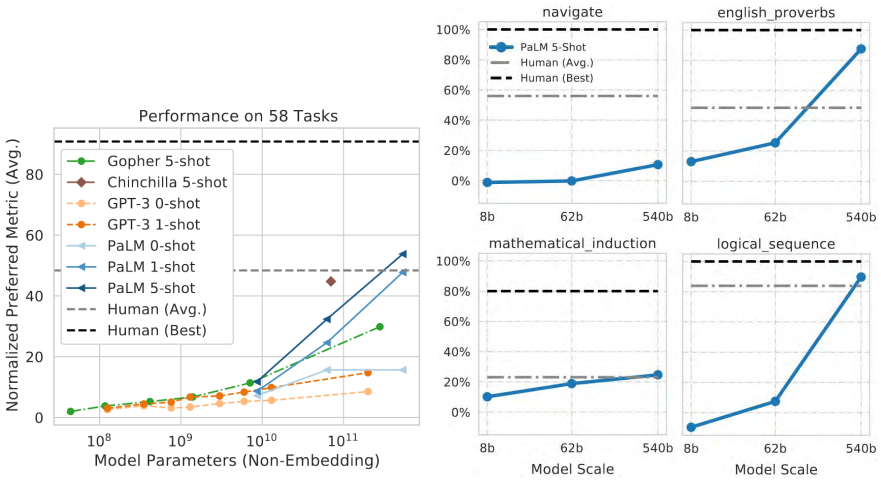


图 3.3 PaLM、GPT-3、Gopher 和 Chinchilla 的评估（左）。此前的模型仅在部分任务上接受评估，因此该图展示了三个模型均参与评估的 58 项任务的汇总结果 [35]。PaLM 的中等准确率高于人类的平均性能。右侧展示了四项特定 BIG任务的结果。 [35, 第 15 页及后续页面]详细比较了三个不同规模的 PaLM 模型与人类水平的性能。

BLEU，而最先进水平为 45.6。对于德英翻译，PaLM 540B 少样本模型的 BLEU 分数达到 47.5，而最先进水平为 45.6。对于摘要生成和问答等其他任务，PaLM 540B 少样本模型的表现接近微调模型，在少数情况下甚至优于它们。

对于语言模型而言，涉及多个中间步骤的推理一直很困难。近期提出了思维链提示（第 3.6.4 节），将中间推理步骤 [226] 加入少样本提示中（图 3.4）。按照这一方法，PaLM 模型在给出最终回答之前，也会针对多步问题生成自己的中间步骤。这提升了模型在若干基准任务上的性能。借助这一技术，PaLM 甚至能够解释笑话，如图 3.5 所示。

3.1.3 Transformer 编码器-解码器

Transformer 编码器-解码器 [212] 通过翻译任务进行了预训练（第 2.3 节）。为提升性能，人们提出了多种替代方案：

- MASS、BART 和 PEGASUS 提出了不同的目标，用于还原受损的预训练数据。例如，预测掩码片段、排列打乱顺序的句子，或插入被省略的词元。

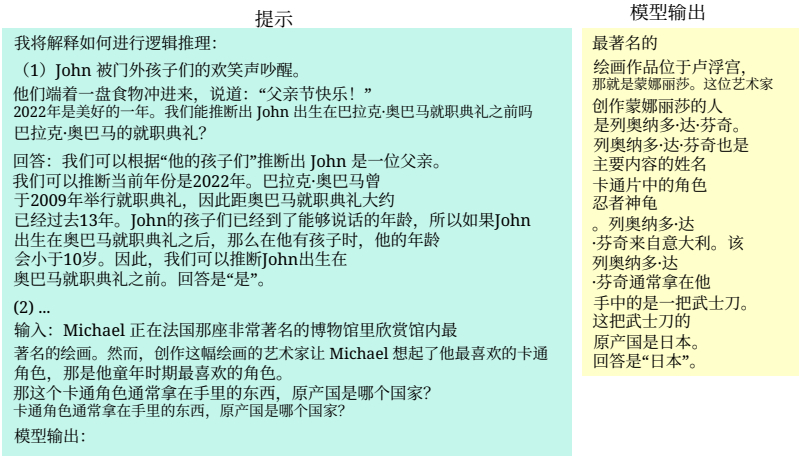


图 3.4 用于常识问答任务的思维链提示的少样本示例 [35, 第 38] 页。将相同的两个思维链示例与要求给出回答的不同提示相结合。



图 3.5 通过使用思维链提示，PaLM 可以解释笑话 [35]。

- T5 将许多语言理解和语言生成任务表述为文本翻译，并使用同一个模型来处理这些任务。
- Longformer、Reformer 和 Transformerl-XL 扩展了输入文本的长度，而不增加参数数量。它们将在第 3.2 节中讨论。

详细信息将在以下段落中给出。表格 3.5 列出了具有代表性的 Transformer 编码器-解码器。

MASS [196] 基于 Transformer 架构。与原始 Transformer 不同，编码器中的一段连续词元会被掩码，而解码器的任务是递归地预测这些掩码词元（图3.6）。因此，MASS 可以联合训练编码器和解码器，使其具备提取嵌入和进行语言建模的能力。MASS 在神经机器翻译、摘要生成和对话

表格 3.5 Transformer 编码器-解码器。预训练和微调损失函数定义见表格 3.1。基准测试：英德 WMT2014 英译德 BLEU，GLUE 第 4.1.1 节准确率，SuperGLUE 第 4.1.2 节准确率，TriviaQA [99] 第 6.2.1 节准确率，宾夕法尼亚树库 [136] 困惑度。基准测试结果仅供参考，因为它们取决于参数数量和计算开销。

模型	预训练部分		微调额外项		基准测试
Transformer [212]	2.3	S2S	S2S	预测翻译后的词元	英德 26.4
UniLM [8]	3.1.3	掩码语言模型、语言模型	多项选择、语言模型	单向和双向	GLUE 87.3
MASS [196]	3.1.3	S2S	S2S	预测掩码词元	英德 28.3
BART [119]	3.1.3	DAE	多项选择、语言模型，S2S	还原损坏文本	GLUE 88.4
T5 [170]	3.1.3	S2S	多项选择、语言模型，S2S	解决许多自然语言处理任务，例如序列到序列问题	GLUE 89.7
GLM [54]	3.1.3	LM	LM	通过以下方式解决所有任务自回归预测	SuperGLUE 82.9
Longformer [10]	3.2.1	掩码语言模型、序列到序列	语言模型、多项选择、S2S	稀疏注意力机制	TriviaQA 77.3
Reformer [108]	3.2.2	语言模型、序列到序列	语言模型、多项选择、S2S	局部敏感哈希，可逆残差层	英德 29.1
Transformer-XL [44]	3.2.2	掩码语言模型，序列到序列	多项选择，序列到序列	稀疏注意力机制	Penn-树河岸 54.5



图3.6 Transformer用于恢复损坏文本的不同预训练任务。跨度掩码是MASS的任务 [196]。BART使用从词元掩码到文档旋转的所有任务 [119]。

回复生成等语言生成任务上进行微调。与先前的 Transformer 架构相比，其性能有显著提升。

BART [119] 采用标准Transformer编码器-解码器架构。预训练任务是恢复经过多种不同方式破坏的文本（图3.6）：像BERT一样预测掩码词元；预测删除的词元及其位置；预测被单个掩码替换的缺失词元；像XLNet一样重建置换句子；以及找出旋转文档的开头。BART经过了GLUE、SQuAD、摘要生成等多项任务的微调，

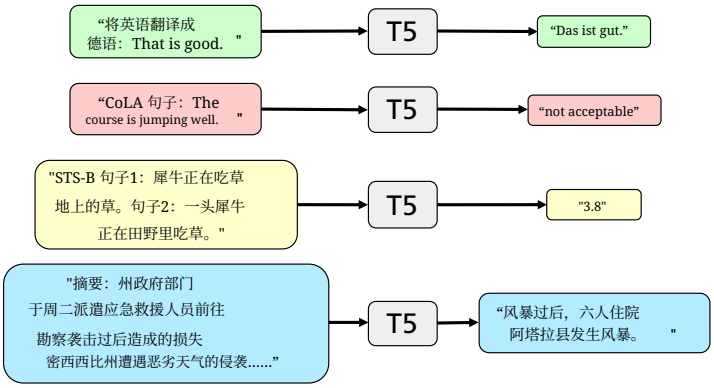


图 3.7 T5 中的每项任务都被表述为翻译任务，其中任务类型以输入文本的前缀形式出现（左侧），模型则生成相应的输出（右侧）。经作者慷慨许可，改编自 [170, 第3] 页。

以及机器翻译。BART在预测被单个掩码替换的缺失词元方面表现最佳。大型BART模型使用与RoBERTa相似的数据集进行训练，其隐藏维度为1,024，包含12层编码器和解码器。其性能与RoBERTa相近。在抽象式摘要任务中，例如在*CNN/Daily Mail* 基准 [78], 上，BART达到了最先进水平。

PEGASUS [251] 提出了一种新的目标，用于在海量文本语料库上预训练大型基于 Transformer 的序列到序列模型：间隔句生成，即掩码或删除句子而非词元。模型必须将这些经过修改的部分生成为一个句子输出。在 12 项文档摘要任务上，该模型取得了最先进水平的性能。

这5 [170] 基于标准Transformer架构。预训练是在一个庞大的训练集上进行的，通过还原损坏文本，将其表述为序列到序列任务。对图3.6中列出的不同预训练任务进行比较后发现，与 BART 类似，文本填充取得了最佳结果。如果原文是“上周谢谢你邀请我参加你的派对。”，模型接收的输入为“谢谢你 [*X*]见面你的派对 [*Y*]周。”，其中包含掩码短语；模型必须生成输出“[*X*] 邀请 [*Y*]上 [*Z*]”，以重建这些掩码短语。

显著跨度掩码 [72] 效果尤为显著。为了聚焦于相关短语，我们训练了 BERT 标注器来识别命名实体（人名、地点等，见第 2.1.3 节），并通过正则表达式识别日期。如果模型需要重建这些跨度，模型性能便会显著提升。通过预测省略的词元，模型能够获取大量句法和语义知识。大量比较表明，序列到序列架构比其他架构（例如自回归语言模型）效果更好。

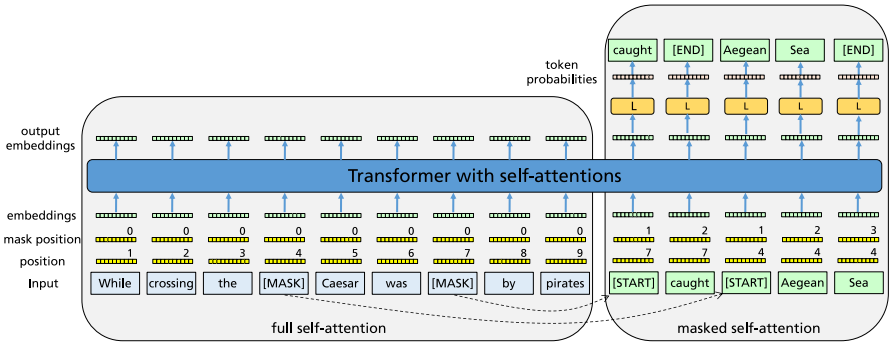


图 3.8 在预训练期间，GLM 的任务是重建被掩码的单个单词或多个单词组成的短语。生成词语在文本和掩码中的位置由位置嵌入指示，这些嵌入会加到词元嵌入上。生成的答案以一个 [结束] 词元 [54]终止。

T5 使用包含 750 GB 清理后英文网页文本的训练数据集，在无监督任务和有监督任务组成的多任务混合上进行预训练。其最大版本有 24 层、128 个注意力头和 110 亿个参数。对于每项任务，数据都会转换为文本到文本格式（图 3.7）。该模型在许多基准测试中取得最先进水平的结果，例如摘要生成、问答、文本分类等。GLUE 的结果为 90.3% [11]。

Primer [195]提出了对原始自注意力架构的两项改进。首先，将 ReLU 激活函数平方。此外，在查询 Q、键 K和值 V的每个多头投影之后，都添加一个卷积层。对于原始 T5 架构，这使训练成本降低了 4 倍。

UniLM2 [8]同时预训练双向语言模型和用于语言生成的序列到序列模型。这两个任务共享模型参数，并重复利用上下文词元的编码结果。该模型使用两种掩码类型：一种是类似于 BERT 的双向掩码，另一种是用于语言建模的伪掩码。借助特殊的自注意力掩码和位置嵌入，该模型可以在一次前向传播中执行两种语言建模任务，而无需对上下文进行冗余计算。在 SQuAD 1.1 阅读理解任务中，该模型的表现优于 $\text{BART}_{\text{BASE}}$；在 CNN/Daily Mail 抽象式摘要任务中，其表现优于 T5_{BASE}。

GLM （通用语言模型） [54, 55] 是 UniLM2 的后继模型，旨在结合 BERT、GPT 和 Transformer 的不同学习范式。在预训练过程中，GLM 的任务是以自回归方式生成多个文本片段，基本采用 GPT 架构。从输入文本 $\boldsymbol{x} = (x_1, \ldots, x_T)$ 中采样出若干个 m 跨度 $x_{i_1}, \ldots, x_{i_1+l_i}$。每个跨度都被替换为一个单独的 [*MASK*]词元，从而得到受损输入 $\boldsymbol{x}_{\text{corrupt}}$。随后，模型依次生成各个跨度中的词元，同时可以访问受损输入以及各跨度中已经生成的词元（图 3.8）。在输入文本中，所有词元

都通过自注意力相互连接，而在输出部分则使用掩码自注意力。每个跨度都以一个 [*END*] 词元结束。为了识别生成词元的位置，会通过嵌入对两个位置进行编码：输入位置和跨度内的位置。需要注意的是，掩码预测可以按任意序列进行，模型必须在重建过程中预测跨度的长度。

微调时，文本分类任务会转换为词预测。为了在情感分类任务中评估句子“服务员们很友善。”，输入会扩展为“服务员们很友善。这真的 [掩码]。”其中，[掩码]必须替换为“好”或“坏”。对于文本生成任务，会在输入文本后附加一个 [掩码] 词元。然后，模型以自回归方式生成续写，作为输出文本。与 BERT 不同，模型能够观察掩码词元之间的依赖关系，从而产生更加一致的预测。与 XLNet 相比，它无需为位置编码额外执行注意力运算，因此降低了计算需求。与 T5 相比，GLM 可以按任意顺序预测跨度，并且所需的额外词元更少。

为了评估模型性能，Du 等人 [54] 使用与 $\text{BERT}_{\text{BASE}}$和 $\text{BERT}_{\text{LARGE}}$相同的训练数据和参数数量（1.1 亿和 3.4 亿）训练了 GLM_{BASE} 和 $\text{GLM}_{\text{LARGE}}$。在这两种模型配置下，GLM 在 SuperGLUE（第 4.1.2 节）上的表现均优于 BERT，例如，$\text{GLM}_{\text{LARGE}}$ 的平均得分为 77.0，而 $\text{BERT}_{\text{LARGE}}$为 72.0。对于一个与 RoBERTa 规模相同的模型，使用更大的预训练数据集时，其 SuperGLUE 平均得分为 82.9，而 RoBERTa 为 81.5。他们表明，通过多任务学习，参数数量相同的单个模型可以同时提高自然语言理解（NLU）的准确率、根据输入生成文本的能力，并解决摘要生成等其他任务 [53]。

更大的模型，如**GLaM** [51] 和**WuDao-2**。**0** [257] 采用混合专家架构，相关内容将在第3.5.2节中介绍。

3.1.4 Transformer变体的系统比较

作为公平比较架构特征的示例，我们报告了以下对预训练语言模型的实验分析，其中Narang等人 [142] 评估了多种Transformer修改的影响。研究考察了以下Transformer特征：

- 激活函数： 除了前馈层中的ReLU激活函数外，还评估了另外11种激活函数。
- 归一化： 除了原始的层归一化外，还探究了五种不同的正则化技术。
- 层数：层数在 6 到 24 之间变化。为确保比较公平，通过调整注意力头的数量以及内部嵌入的宽度 $d$ff，使参数数量保持不变。

• 词元嵌入： 我们将原始 Transformer 嵌入与五种分解嵌入变体进行了比较。此外，还研究了 Transformer 块的共享情况。

• *Softmax*： 研究将用于计算词元概率的标准 Softmax 与三种 Softmax 变体进行了比较。

• 架构： 作者将基础 Transformer 与另外 17 种架构进行了比较。在大多数情况下，参数数量基本保持一致。

作者在两种设置下评估了这些变体：基于 T5 Transformer 的迁移学习（第 3.1.3 节），以及在 *WMT2014* 英德翻译任务 [17]上进行的监督机器翻译。谨慎而言，这些结果也可应用于 BERT 和 GPT 等其他类型的预训练语言模型。

T5 的每种架构变体都使用“跨度破坏”掩码语言建模目标，在由 806 GB 数据组成的 *C4*数据集 [171] 上进行预训练。随后，T5 在三个任务上进行了微调：*SuperGLUE* 语言理解任务 [219]、*XSum* 抽象式摘要数据集 [143]、以及 *WebQuestions* 基准测试[13]，其中没有提供额外知识作为背景信息。每个模型的计算开销和参数数量都固定在相同水平。一个例外是参数数量显著较少的架构，它接受了更长时间的训练。

多种激活函数的性能优于 ReLU 激活，尤其是*SwiGLU* 和*GEGLU*。它们是门控线性单元（GLU），会与另一种激活函数形成乘积 [189]。无论是预训练、微调还是监督训练，性能都有所提升，且不会影响计算时间。例如，在 SuperGLUE 上，得分从 71.7% 提升至约 76.0%。将层归一化替换为 *RMS*归一化 [249] ，可以提升所有任务的性能。例如，SuperGLUE 得分从 71.7% 提升至 75.5%。此外，训练速度也更快。

正如预期的那样，在参数数量保持不变的情况下，增加模型深度通常会带来更好的性能。在 SuperGLUE 上，具有 18 层的模型得分达到 76.5%，而基础模型为 71.7%。WebQuestions 和翻译任务也有类似的提升，但摘要生成任务没有提升。这与理论结果一致（第 3.5.1 节）。缺点是，深度更大的模型需要更多计算时间。

在不同层之间共享参数的架构通常会导致性能下降。对编码器和解码器使用相同嵌入的效果尚不明确。将嵌入因子分解为矩阵乘积通常会导致较差的结果。如果使用 *Softmax*混合 [239] 来预测输出概率，性能通常会更好，例如 SuperGLUE 的得分提高到 76.8%。不过，这种方法需要多达 40% 的额外计算开销。

在评估的架构变体中，Synthesizers 与点积注意力（第 3.2.2 节）的两种组合表现优于标准 Transformer。Synthesizers 不会计算嵌入之间的“相关性”，而是根据单个嵌入或随机确定注意力权重。Switch Transformer、专家混合和乘积键记忆的参数都显著多于

基线 Transformer，但能够提升性能。Switch Transformer（[56] 第 3.5.2 节）的参数远多于基础 T5模型。要达到与 Switch 相同的性能，T5 需要多用七倍的训练 FLOPS（每秒浮点运算次数）。专家混合模型 [116] 将编码器和解码器中的计算分配给 2 个专家模型。乘积键记忆（[112]第 3.1.1 节）以最近邻搜索取代点积注意力。

对于其他 12 种架构，它们的表现均未优于标准 Transformer [142]。这与提出这些模型的论文中的发现不同。原因似乎是，Transformer 架构的改动难以迁移到其他代码库和应用中。因此，作者建议在不同的底层实现中尝试新的改动。此外，应在多种下游应用中评估新方法，包括迁移学习、监督学习和语言建模。应固定超参数 优化，以确保方法的稳健性。最后，应报告结果的均值和标准差，以避免只挑选单个最佳结果。

3.1.5 总结

预训练任务的改动对预训练语言模型的性能有着深远影响。人们评估了许多不同类型的预训练损失，例如掩码短语预测、替换词元检测或句子顺序识别。根据基准测试，XLNET 对置换词元的预测尤其有益，因为 XLNET 考虑到了掩码词元之间的依赖关系。此外，DeBERTa 的解耦的词元和位置嵌入能够提升下游分类器的性能。在应用方面，像 BERT 这样的自编码器对第5章中的信息抽取尤为重要。

对于 GPT 等自回归预训练语言模型，已有多种模型规模更大、训练数据更多的变体问世。然而，在大多数情况下，预训练任务并未改变。训练更大的模型需要改进并行计算基础设施，并使文本生成性能达到前所未有的水平。通过创建自定义起始文本（提示），这些模型无需进一步微调（第 3.6.3 节），就能以极高的准确率完成大量特定任务。预训练语言模型所捕获的知识在数量和质量上都出乎意料地高，第4章将对此进行讨论。在应用方面，自回归预训练语言模型尤其用于文本生成（第 6 章）和图像生成（第 7.2 节）。由于用途广泛且性能大幅提升，近年来的大规模预训练语言模型被称为基础模型。

编码器-解码器 Transformer 最初用于将文本从一种语言翻译成另一种语言。针对这些模型，人们评估了许多新的预训练任务。其中一些与自编码器的任务类似，例如预测被掩码的跨度或插入省略的词元。其他任务则根据输入-输出架

构进行了调整，例如重建句子排列和文档旋转。在 GLUE 和 SuperGLUE 自然语言理解任务中，BART 和 T5 的表现最佳。通过创建额外的合成训练样本，T5 和其他模型的性能可以得到提升（第 3.6.6 节）。

对 Transformer 架构进行的系统性比较表明，若干架构变更提升了性能。与 ReLU 相比，采用 SwiGLU 和 GEGLU 激活函数使 SuperGLUE 的准确率提高了 4% 以上。使用 RMS归一化取代层归一化，也观察到了类似的提升。即使参数数量保持不变，增加模型深度也能带来更好的性能。以 k-means 聚类取代标量积的 Synthesizers、专家混合和乘积键，表现也优于标准Transformer。

T5 和 GLM 表明，通过指令提示控制的 Transformer 可用于解决文本分类、文本生成和文本翻译中的任意问题。因此，它们结合了 BERT、GPT 和翻译模型的能力。Transformer 被广泛用于复杂的文本生成任务，例如机器翻译（第 6.3 节）、对话（第 6.6 节）和图像生成（第 7.2 节）。

3.2 捕获更长依赖关系

众所周知，自注意力的一个问题是其时间和内存复杂度呈二次增长，这在许多场景下会妨碍模型的可扩展性（第 2.1.6 节）。如果序列长度 T 增加到 $2T$，那么就必须计算词元之间数量多出四倍的关联（注意力）。这限制了模型在任务需要更大上下文时的直接应用，例如回答问题或总结文档。此外，训练时还需要更多内存来存储注意力。因此，人们提出了许多概念，以在不过度增加计算和内存需求的情况下处理长序列。

- BigBird、Sparse Transformer、Longformer 和 GPT-3 使用稀疏注意力矩阵来减少参数数量。
- Reformer 通过局部敏感哈希对词元进行聚类，从而减少需要计算的注意力数量。
- Performer 和线性Transformer通过对注意力矩阵进行低秩近似，或采用基于核的自注意力形式，来减少参数数量。
- Transformer-XL 和 LinearTransformer 以自回归方式重用之前文本片段中的计算结果，从而降低计算开销。

[59, 207]提供了扩大输入序列的技术综述。

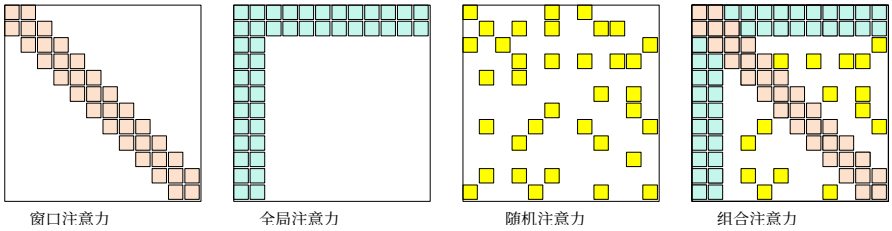


图 3.9 BigBird 中用于计算输入词元之间关联的注意力机制 [247]。该矩阵表示词元对之间的注意力：序列邻居之间的注意力（左）、对少数词元的全局注意力（左起第二幅）、随机注意力（左起第三幅）以及组合后的 BigBird 注意力（右）。白色块表示省略的注意力对。

3.2.1 稀疏注意力矩阵

BigBird [247] 通过按照预先设定的模式省略注意力关系矩阵中的条目，减少注意力计算次数。BigBird 扩展了基于Transformer的模型，例如 BERT，并使用一组 g 全局标记，这些标记会关注序列中的所有词元。此外，每个词元 v_t 会关注一组 n_l 局部相邻词元以及一组 n_r 随机词元。由此得到的关联矩阵如图 3.9 所示。如果数字 g、n_l 和 n_r 不会随着序列长度 T 增加，那么注意力次数将随 T线性增长。

该模型的构造方式使得任意词元对之间沿中间词元的路径长度都保持较小，就像小世界图一样。作者证明，他们的模型仅使用 $O(T)$ 个内积，就能够表达所有连续的序列到序列函数（表3.6）。此外，他们还表明，在标准假设下，BigBird 是图灵完备的，即能够执行任意计算（另见 [246]）。BigBird 注意力模块可用于 BERT、自回归语言模型和 Transformer架构。在许多应用中，使用序列长度 4,096 的 BigBird 能够提升最先进水平，例如在需要根据给定证据进行多跳推理的问答任务中。需要注意的是，在一组实验中，不使用随机注意力的 BigBird 表现优于使用随机注意力的 BigBird。

采用这些概念的早期模型包括 *Sparse Transformer* [33] 和*Longformer* [10]，，它们与 WaveNet [148] 类似，采用步幅式或“扩张式”邻域。在这里，并非所有相邻邻居都会受到一个词元的关注，而是仅关注每隔 d 个的邻居，其中 $d > 1$。如果使用 k 层，该构造可覆盖 d^k 个邻居，从而实现跨越较大距离的关联。**Extended Trans-former Construction**（ETC）模型 [3] 将全局标记的概念推广开来，全局标记能够在整个序列中与相距较远的词元进行关联通信。

GPT-3 [25]（第 3.1.2 节）是一种近期的语言模型，拥有 96 层、96 个注意力头和 1750 亿个参数，可处理长度为 2,048 的序列。为应对过长的序列，作者采用了“在 Transformer 的各层中交替使用密集注意力模式与局部带状

表3.6 用于建模长距离依赖的稀疏自注意力模型。T 为序列长度，g 为全局标记数量，k 为窗口大小。（参见 [207]）

模型						
Transformer-XL [44]	T^2	-	X	-	-	-
Reformer [108]	$T \log T$	-	-	-	-	X
Routing Transformer [180]	T 对数 T	-	-	X	-	X
压缩 Transformer [169]	T^2	-	X	X	-	-
ETC [3]	$g^2 + Tg$	-	-	X	X	-
GPT-3 [25]	$T\sqrt{T}$	-	-	-	X	-
Performer [34]	T	X	-	-	-	-
线性Transformer [105]	T	X	-	-	-	-
BigBird [247]	T	-	-	X	X	-
S4 [68]	T	X	-	-	-	-

稀疏注意力模式，类似于 Sparse Transformer" [33]。该架构的详细信息目前尚不清楚。该模型在语言建模、问答等方面取得了前所未有的性能，相关内容将在第 3.6.3 节讨论。

3.2.2 哈希与低秩近似

Reformer [108] 引入了局部敏感哈希，用于将具有相似键/查询向量的词元聚类到一起。这种方法会以较高的概率将相似的输入项哈希到同一个"桶"中。每个簇使用相同的查询/键参数。通过这种方式，词元以数据驱动的方式聚合起来。类似地，*RoutingTransformer* [180]通过 k-均值聚类对词元进行聚类。

Transformer-XL [44]会重复利用序列先前片段的计算结果。将这种递归机制应用于语料库中每两个连续的文本片段后，它实际上在隐藏状态中建立了片段级递归。借助多层结构，实际使用的上下文可以远远超出两个文本片段的范围。压缩 *Transformer* [169]也采用了类似的方法。*Segatron* 是一种变体，它将文档中的段落索引、段落中的句子索引以及句子中的词元索引编码为嵌入，并将其添加到词元嵌入中。这一修改改善了语言建模中的困惑度。

该 **Performer** [34]通过对自注意力矩阵进行低秩近似，降低了计算负担。它使用具有正交随机特征的随机核来计算自注意力。借助正交性，作者无需计算完整的乘积方阵，因为正交特征的点积为 0。因此，计算需求随序列长度线性增长。作者能够证明，他们的模型几乎可以实现——

对完整注意力矩阵进行无偏估计，同时保证近似结果具有一致收敛性和更低的方差。

这款 **线性Transformer** [105] 也采用基于核的形式来实现自注意力，将复杂度降至线性。为了根据过去的输入预测未来元素，作者构造了一种类似于循环神经网络的迭代算法，其速度远快于标准Transformer。研究表明，该模型在预测性能损失不大的情况下，可将推理速度提升最多三个数量级。

Transformer-LS（长短 Transformer） [258] 在相邻词元之间采用局部滑动窗口注意力，并通过动态投影实现长距离注意力，以表示相距较远词元之间的关系。动态低秩投影取决于输入序列的内容。作者声称，该方法对于插入、删除、改述等操作具有更强的鲁棒性。该方案在不同基准测试的语言建模任务上取得了最先进的困惑度，例如 enwik8 上达到 0.99；在 ImageNet 上，也取得了视觉Transformer领域的最先进结果。

Combiner [174] 通过键向量表示嵌入组。给定词元 v_t 关注词元 v_s 的概率由一个乘积表示：v_t 首先关注代表一组地点的键向量，该组地点包含 v_s；然后将其与在该组中选择 v_s 的概率相乘。通过这种方式，Combiner 可应用于长度最长达 12,000 的序列。该方法在大型基准测试上实现了最先进的困惑度。此外，它还提升了 长程竞技场基准 [209]上的平均性能，该基准专门用于评估模型处理长文档的质量。

Synthesizer [206]用“合成函数”取代注意力中的成对点积。这些函数能够学习注意力矩阵，而这些矩阵可能依赖于输入词元，也可能不依赖（参见第 3.1.4 节）。在 Dense Synthesizer 中，层内每个词元嵌入 $x_i, i = 1, \ldots, T$都会经过一个带 ReLU 激活的两层非线性前馈网络，投影为长度为 T 的向量。该向量的值用作权重，以确定各值的混合比例，从而形成输出嵌入。因此，它不会像标准自注意力那样计算嵌入之间的“相关性”来确定它们的相似度。还有一种极端变体，其混合比例是随机设定的。尽管如此，在机器翻译、语言建模、对话生成、掩码语言建模和文档分类等多项任务上，这种“合成”注意力的性能仍可与普通自注意力相媲美。将 Random Synthesizers 与常规点积注意力结合起来，在若干基准测试上能够胜过 T5。

Perceiver [92] 定义了一种非对称注意力机制，通过迭代将长输入序列 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ （例如图像中的 50k 个像素）转换为较短的潜在单元序列 $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ （例如 $n = 512$），形成一道瓶颈，输入必须经过这道瓶颈（图 3.10）。通过交叉注意力（第 2.3.1 节），经过 Q转换的潜在序列嵌入 $Q\boldsymbol{u}_i$ 与经过 K转换的长输入序列嵌入 $K\boldsymbol{x}_j$形成一个标量积（$Q\boldsymbol{u}_i$）（$K\boldsymbol{x}_j$）。该标量积被用作经过 V转换的长序列嵌入 $V\boldsymbol{x}_j$的权重，以生成新的短嵌入。Perceiver 本质上是一个 BERT模型，只是序列长度为

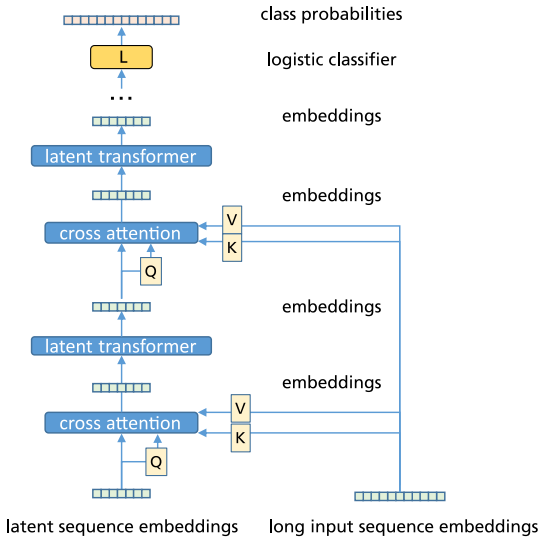


图 3.10 如果输入序列过长，Perceiver 会定义一个较短的潜在序列。通过长序列与潜在序列之间的交叉注意力，信息得以压缩。标准 Transformer块会计算潜在序列元素之间的自注意力，最终生成一个分类结果 [92]。

n 而不是 T，从而避免计算开销随输入长度呈二次方增长。这种迭代方法使模型能够将有限的容量用于最相关的输入。在实验中，Perceiver 在图像分类方面的表现优于领先的 ResNet-50卷积神经网络 [92]。*Perceiver IO* [93]通过另一次交叉注意力运算，将 Perceiver 得到的 n 输出嵌入投射为更长的输出嵌入序列；例如，该运算会将输出元素的位置嵌入作为查询向量。*Perceiver AR* [73]扩展了 Perceiver，使其能够像编码器-解码器 Transformer 一样顺序生成输出。

S4 [68] 是一种基于卡尔曼滤波器的结构化状态空间序列模型，用于观测存在错误的状态模型 [101]。连续状态空间模型定义如下：

$$\boldsymbol{x}'(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \qquad \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{D}\boldsymbol{u}(t), \tag{3.1}$$

它通过潜在状态 $\boldsymbol{x}(t)$ 将输入信号 $\boldsymbol{u}(t)$ 映射为输出 $\boldsymbol{y}(t)$。作者对矩阵 $\boldsymbol{A}$ 重新参数化，并将其分解为低秩项与斜对称项之和。此外，他们还计算了相关无限序列的生成函数，并将其在频域截断至某个长度 L。低秩项可以通过用于矩阵求逆的 Woodbury恒等式进行校正。斜对称项可以对角化，并可化简为柯西核 [153]。

$\boldsymbol{A}$ 矩阵通过一个特殊的上三角“HiPPO”矩阵进行初始化，该矩阵使状态 $\boldsymbol{x}(t)$ 能够记忆输入 $\boldsymbol{u}(t)$ 的历史。作者证明，在复数空间 $\mathbb{C}$ 中，相应的状态空间模型可以表示为矩阵 $(\boldsymbol{\Lambda} - \boldsymbol{P}\boldsymbol{Q}^*, \boldsymbol{B}, \boldsymbol{C})$、某个对角矩阵 $\boldsymbol{\Lambda}$ 以及向量 $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{B}$, $\boldsymbol{C} \in \mathbb{C}$。这些是 S4 的可训练参数，其中 N 是状态维度。总的来说，

表格 3.7 长程竞技场基准的准确率结果。最佳得分以粗体显示，优于标准Transformer的结果带有下划线。（参见 [209]）

模型	ListOps	文本分类	检索	图像分类	Pathfinder	Path-X	平均值
Transformer	36.3	64.3	57.5	42.4	71.4	×	54.4
Reformer	37.3	56.1	53.4	38.1	68.5	×	50.7
Synthesizer	37.0	61.9	54.7	41.6	69.5	×	52.9
BigBird	36.0	64.0	59.3	40.8	74.9	×	55.0
线性 Transformer。	16.1	65.9	53.1	42.3	75.3	×	50.6
Performer	18.0	65.4	53.8	42.8	77.0	×	51.4
S4	**58.4**	**76.0**	**87.1**	**87.3**	**86.1**	**88.1**	**80.5**

S4 定义了一种形状为（批量大小、序列长度、隐藏维度）的序列到序列映射，其方式与 Transformer、循环神经网络和卷积神经网络等相关序列模型相同。对于序列长度 L，这需要 $O(N+L)$ 的计算开销和 $O(N+L)$ 的内存空间，这接近序列模型所需的最低值。Gu等人 [69]对 S4 模型进行了详细阐述并提供了实现。

实证评估结果表明，对于长度为 1,024 的输入，S4 的速度比标准 Transformer快 1.6 倍，且仅需其 43% 的内存。对于长度为 4096 的输入，S4 的速度快 5 倍，且仅需标准Transformer 9% 的内存。在*Long Range Arena*基准测试的基准测试中，S4 将最先进水平的平均准确率从 59.4% 提高到 80.5%（表 3.7）。此外，S4 成功解决了极具挑战性的 Path-X 任务，该任务需要对长度为 16k 的序列进行推理，而此前所有模型都未能做到这一点。最后，S4 能够对长度为 16k 的序列进行原始语音信号分类，并以 98.3% 的准确率创下新的最先进水平。S4 在长程序列处理方面实现了真正的突破。此外，S4 在长程时间序列预测方面表现更佳，例如，在预测 30 天的天气数据时，均方误差降低了 37%。*DSS* [70] 是 S4 的一个变体，形式更简单，但性能略低。

3.2.3 长输入序列下 Transformer 的比较

Long Range Arena [209] 旨在评估处理长度从 1k 到 16k 词元的长输入序列任务时的性能。它包含六个不同的基准数据集，涵盖文本、图像、数学表达式和视觉空间推理。任务包括 ListOps（使用列表表示法进行计算）、文本分类（使用字符序列对 IMDB影评进行分类）、文档检索（基于文档嵌入）、图像分类（基于像素序列）以及两个版本的 Pathfinder（检测圆形）。作者评估了九种能够处理长输入的 Transformer 架构。

结果如表 3.7 所示。对于 ListOps 的层次结构化数据，事实证明，基于核的方法（例如 Performer 和

线性Transformer）并不适用。对于文本分类，基于核的方法表现尤其出色。对于图像分类，除 Reformer 外，大多数模型都表现良好。所有模型都以可接受的性能解决了寻路任务，其中 Performer 的表现最佳。然而，除 S4 外，所有模型都无法完成扩展寻路任务，也无法找到解决方案。就所有基准测试而言，S4 以较大间隔成为最佳模型。

在速度方面，Performer 表现最佳，在长度为 4k 的序列上比标准Transformer快 5.7 倍。内存消耗从标准Transformer的 9.5 GB 到线性Transformer的约 1.1 GB 不等。除 Synthesizer 外，其他所有模型的内存消耗都不到 3 GB，其中 S4 在这两个方面都表现出色。

3.2.4 总结

针对预训练语言模型如何高效处理长输入序列，已有多种方案。通常会采用稀疏注意力矩阵，只使用部分可能的注意力来建立远距离位置之间的联系，而近距离位置通常采用完全注意力计算。一些词元采用全局注意力，以便在未直接连接的位置之间传递信息。BigBird 是一个著名示例，它加入了随机注意力。其计算开销仅随输入大小线性增长，同时仍能执行任意序列计算。Performer 和线性Transformer 等其他架构也呈现线性增长。

有些架构通过低秩因子分解近似注意力矩阵，或聚合表达相似内容的词元（Reformer、Combiner）。另一种方法是使用递归机制，从而减少对相距较远的词元进行的计算（Transformer-XL、线性Transformer、Transformer-LS、Perceiver）。另一种选择是对自注意力矩阵进行因子分解（Performer），或用更简单的计算取代它（Synthesizer）。最近，有人提出了 S4 模型，将状态空间模型应用于长程预测。它采用基于复数计算的架构，与通常的 Transformer 架构截然不同。该模型的性能远超所有先前模型，优势显著，而且在计算时间和内存方面都很高效。

这些方法的性能通过 Long Range Arena 的六种不同基准测试进行了评估。结果表明，在所有基准测试中，S4 的表现都优于其他模型。与标准Transformer相比，所有方法都能够降低内存消耗。更长的输入长度带来了新的应用，例如原始语音处理、图像处理或基因组学 [247]。

3.3 多语言预训练语言模型

世界上有 7,100 多种语言 [9], , 每种语言几乎都能表达所有事实和概念。因此，预训练语言模型也应该能够为不同语言中的概念生成一致的表示。简单陈述句中的动词、主语和宾语的基本语序在不同语言间存在一定差异。例如，英语、德语、法语和普通话都是主谓宾语序语言（主语-谓语-宾语）[100]。在这类语言中，动词通常位于主语和宾语之间。另一方面，印地语和日语属于主宾谓语序语言，也就是说，动词位于主句末尾。爱尔兰语和阿拉伯语则属于谓主宾语序语言。基本语序相同的两种语言通常还具有其他相似之处。例如，谓宾语序语言通常使用前置词，而宾谓语序语言通常使用后置词。此外，一种语言中可能存在词汇空缺，即没有单词或短语能够准确表达另一种语言中某个单词的意义。一个例子是德语单词“幸灾乐祸”，大致可译为“因为别人倒霉而感到高兴”。 [100]还讨论了更多此类差异。

要实现跨语言理解，预训练语言模型必须使用不止一种语言进行训练，并且必须捕捉这些语言的结构差异。在训练过程中，预训练语言模型可以建立不同语言中概念之间的对齐关系。

- 在多语言数据上使用共享词元词表训练大型预训练语言模型，例如 T5 或 BERT，可以得到通过利用语言之间的共同结构来实现信息跨语言转移的模型。
- 类 BERT 模型可以通过掩码语言建模进行训练，使一种语言中句子的词语与其翻译成另一种语言后的词语建立关联。然而，研究表明，即使只有很少或完全没有平行训练数据，也可以进行多语言处理。
- Transformer 编码器-解码器模型经过明确训练，可将文本从一种语言翻译成另一种语言。

并行使用多种语言训练语言模型可以提升性能，尤其是对于训练数据较少的语言。这一点已经在静态单词嵌入方面得到证明 [194]。

3.3.1 自编码器模型

mBERT（多语言BERT）[48] 是一种标准的BERT模型。它使用来自104种语言的非平行维基百科文本，通过掩码语言建模损失进行预训练，并为所有语言共享一个包含11万个WordPiece词元的词元词汇表。这意味着中文实际上是按字符进行分词的。每个训练样本都是一种语言的文档，且不存在跨语言词典或训练标准。为展示其性质，该模型经过微调，应用于多语言版本的*XNLI*[40] 自然语言推理（NLI）基准测试，即预测以下任务：

表 3.8 6 种语言的跨语言自然语言推理（XNLI） [40] 测试准确率。比较了对所有语言使用 XNLI 进行微调与仅对英语使用 XNLI 进行微调的结果。mBERT [48]和 XLM [111]的结果。

使用.....进行微调	模型	英语	中文	西班牙语	德语	阿拉伯语	乌尔都语	
所有语言	mBERT	81.9	76.6	77.8	75.9	70.7	61.6	
仅英语	mBERT	81.4	63.8	74.3	70.5	62.1	58.3	
所有语言	XLM	85.0	78.6	80.8	80.3	76.5	63.2	
仅英语	XLM	85.0	76.5	78.9	77.8	73.1	57.3	

第一句话是否蕴含第二句话。事实证明，mBERT可以使用单一语言在NLI任务上进行微调，并且在相关语言 [40, 232]上仍能取得良好的测试结果。

6种语言的结果 [111] 如表3.8所示。与使用所有语言对XNLI进行微调相比，如果使用英语XNLI进行微调并在其他语言中进行评估，那么对于相关语言（例如西班牙语和德语），准确率仅略有下降。对于其他语言，性能下降幅度更大，但结果仍然不错。即使对于使用不同文字的语言之间，也存在信息转移，例如阿拉伯语和乌尔都语。作者还考察了一个单词及其翻译的嵌入。结果表明，尽管语言之间不存在对齐，但一个单词及其翻译之间的余弦相似度为0.55。

Karthikeyan 等人 [104] 研究了 mBERT 成功的因素。他们发现，即使词元词汇表之间完全没有重叠，mBERT 仍具备跨语言能力。此外，两种词汇表中相同词元数量的增加对性能提升贡献甚微。在比较不同语言对时，作者表明，双语 BERT 的较大网络深度和较高的参数总量对于单语和跨语言性能都至关重要，而注意力头的数量并不是显著因素。另一方面，源语言和目标语言的结构相似性，即词序和词语频率，对跨语言性能有很大影响。

XLM [111]通过在预训练期间使用来自不同语言对的翻译句子，改善了不同语言之间的知识转移。作者将一个句子及其另一种语言的翻译拼接起来进行训练，并引入了新的翻译语言建模 （*TLM*）目标，以改进跨语言预训练。为了预测输入句子中的掩码词，算法可以关注翻译句子中的词语。通过这种方式，模型学会关联不同语言中的词语。图 3.11 展示了一个示例。如表 3.8 所示，与 mBERT 相比，XLM 在 XNLI 上具有高得多的跨语言准确率。从在英语上微调的模型向其他语言转移时，仅产生了很小的损失。实验表明，TLM 能够使 XNLI 的准确率平均提高 3.6%。研究人员还评估了该模型从德语和其他语言到英语的无监督机器翻译性能，结果非常好（参见第 6.3 节）。

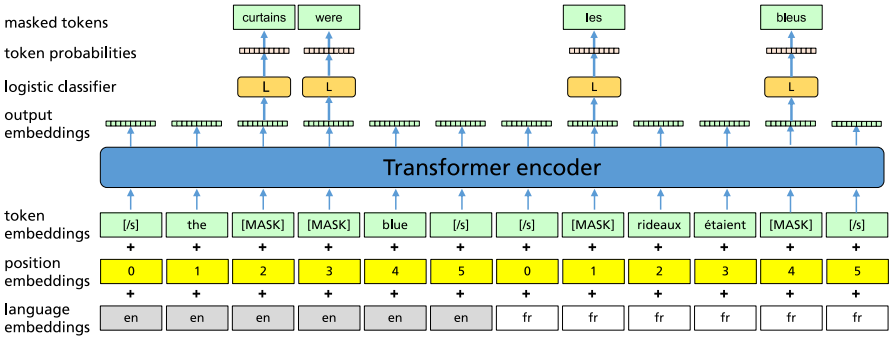


图 3.11 翻译语言建模（TLM）任务应用于翻译句子对。为了预测被掩码的英语单词，模型可以同时关注英语句子及其法语翻译，因此能够促进英语和法语表示的对齐 [111]。

Unicoder [88]是改进版的 XLM 模型，增加了三个训练任务。跨语言词对齐学习关联翻译句子中对应的词语。跨语言释义检测将来自不同语言的两个句子作为输入，并判断它们是否具有相同的意义。文档级跨语言掩码语言模型将 MLM 任务应用于文档，其中部分句子被其翻译句替换。在 XNLI 上，作者报告平均准确率提高了 1.8%。

XLM-R 是 XLM 的优化版本 [41]。它基于 RoBERTa 构建，并在一个规模庞大的多语言 CommonCrawl 数据集上进行训练，该数据集大小为 2.5TB，涵盖 100 种语言，并采用包含 25 万个词元的通用词汇表。它将 XNLI 得分的最先进水平提高到 79.2%。对于跨语言问答，模型在英语 SQuAD 数据集上进行微调，并在另外 7 种语言上进行评估。XLM-R 将该 SQuAD 版本的 F1 分数提高了 9.1%，达到 70.7%。在跨语言分类任务中，对于低资源语言，XLM-R 的准确率最高比 mBERT 高出 23%。XLM-R 的性能几乎与强大的单语模型一样出色。

这些结果支持了这样的观察：通过在大量文本上训练，可以提升预训练语言模型的性能 [102]。在模型容量足够大的情况下，增加语言种类有助于提升模型在低资源语言上的跨语言性能。结合 Aghajanyan 等人提出的方法 [2],（该方法可避免微调过程中表示发生过大的变化，见第 3.6 节），$\text{XLM-R}_{\text{LARGE}}$ 模型将 XNLI 的最先进水平提高到了 81.4%。如果在 XLM-R 中加入一项额外标准，将不同语言中语义等价的句子与其他句子区分开来，那么其在语义任务上的准确率会有所提升 [228]。更大的模型（例如拥有 107 亿个参数的$XLM\text{-}R_{XXL}$ [66] ）也在 CC-100 上进行了预训练。CC-100 包含 1670 亿个词元的非平行文本，也涵盖低资源语言；这些模型使 XNLI 性能提升了 2.4%。

RemBERT [37] 重新分配了多语言模型的参数。作者首先表明，在最先进水平的

预训练语言模型中使用不同的输入和输出嵌入，可以提升模型性能。随后，他们证明，为输出嵌入分配更多参数能够提高模型准确率，而且这种提升在微调过程中得以保持。因此，Transformer 表示具有更强的泛化能力，也更易迁移到其他任务和语言。Xtreme</style}集合 [86] 是一个多任务基准，用于评估多语言表示在 40 种语言和 9 项任务上的跨语言泛化能力。尽管 RemBERT 仅使用规模更小的训练数据子集进行训练，并额外涵盖了十种语言，但其表现仍优于 XLM-R。

诸如 BERT 之类的预训练语言模型会生成上下文词元嵌入。然而，用户通常需要句子或段落的上下文<strong style id='1'>嵌入来比较其内容。**LaBSE** [57] 是一种语言无关的段落嵌入生成器，其中源句子和目标句子使用共享的基于BERT的编码器分别进行编码。最终层中 [*CLS*] 的表示被用作每个输入的句子嵌入。LaBSE 将掩码语言模型（MLM）和翻译语言模型（TLM）的损失与间隔准则相结合。该准则计算段落嵌入 $\boldsymbol{x}$ 与其正确翻译的嵌入 $\boldsymbol{y}$ 之间的余弦距离 $\cos(x, y)$。随后要求 $cos(\boldsymbol{x}, \boldsymbol{y}) - m$ 大于 $\cos(\boldsymbol{x}, \boldsymbol{y}_i)$，其中 m 为正间隔， $\boldsymbol{y}_i$ 则是任意其他段落的嵌入。LaBSE 使用170亿个单语句子和60亿个双语翻译句子进行训练。由此得到的句子嵌入显著提高了跨语言信息检索中句子的检索准确率Sota（参见第6.1节）。代码和预训练模型均已提供。

3.3.2 Seq2seq Transformer模型

mT5 是 T5 Seq2seq Transformer（第 3.1.3 节）的多语言版本，参数量最高可达 13B [236]。它使用涵盖 101 种语言、约 480 亿个词元以及包含 25 万个词元的通用词汇表的网页训练数据集进行了预训练。在预训练过程中，该模型必须像 T5 一样，预测单语文档中的掩码短语。与 T5 类似，该模型可以通过前缀（例如“summarize”）被指示执行不同的任务。这些任务通过对相应数据集进行微调进行训练。

对于 *XNLI* 基准 [40]，模型需要判断第一个句子是否蕴含第二个句子。使用英语数据在 XNLI 上对模型进行微调，并在 15 种语言上测量性能时，其准确率为 84.8%；相比之下，mBERT 为 65.4%，XLM 为 69.1%，XLM-R 为 79.2%。尽管不同语言的文本并非平行文本，模型仍能够利用语言之间的结构相似性来完成任务。该模型的代码可在 [235] 获取。类似模型也用于多语言翻译（第 6.3 节）。**mT6** [31]通过使用成对的翻译句子来改进 mT5 的训练，并定义了新的训练任务。实验结果表明，与 mT5 相比，mT6 的跨语言能力有所提升。另一项改进是 **Switch** [56]，它采用了混合专家

（*MoE*）架构的 mT5 只需 mT5 五分之一的训练时间，便能在全部 101 种语言上提升性能（第 3.5.2 节）。

mBART [126] 是基于 BART 模型（第 3.1.3 节）的多语言编码器-解码器。输入文本通过掩码短语和打乱句子顺序进行损坏，然后对单个 Transformer 模型进行预训练，使其恢复损坏文本。这一过程使用涵盖 25 种语言的训练文档完成。随后，使用翻译任务对预训练模型进行微调，该任务涉及单个语言对。此外，还可以使用回译：训练另一个模型，将目标句子翻译回源语言，并通过额外的损失促使模型重构源句子。mBART 会在编码器输入的末尾和解码器输入的开头分别添加语言符号。这使模型能够知道需要编码和生成的语言。事实证明，预训练能够改善翻译效果，尤其是对于平行训练数据较少的语言。此外，回译还显著改善了翻译结果。研究者进行了许多实验，以分析不同算法特征的影响。当翻译完整文档而非单个句子时，预训练尤其重要。

mBART 也可用于无监督机器翻译，此类翻译完全不使用任何平行文本。作者使用预训练权重初始化模型，然后学习根据回译生成的源句子预测单语句子。对于结构相似的语言，结果非常好，例如在英德翻译中，mBART 的 BLEU 分数为 29.8，接近监督方法的 30.9。请注意，mBART 的性能与 MASS 相近（第 3.1.3 节）。对于结构差异较大的语言对，例如英语-尼泊尔语，mBART 的结果尚可，而其他方法则无法奏效。

MARGE [118] 是一种多语言序列到序列模型，通过检索其他语言的文档 $z_1, \ldots, z_k$ 来重建 x 以某种语言写成的文档。该模型使用来自维基百科和 CC-News 的 26 种语言文本进行训练。文档通过 Transformer [212]第一个词元的输出嵌入进行编码。检索模型通过嵌入每篇文档并计算它们的余弦相似度，对目标文档 x 与每篇证据文档 z_j 的相关性 $f(x, z_j)$ 进行评分。Transformer 接收检索得到的 $z_1, \ldots, z_k$ 的嵌入文本以及辅助相关性分数 $f(x, z_j)$ 作为输入，并经过训练后生成目标文档 x 作为输出。相似度分数用于对解码器到编码器的交叉注意力加权，使解码器更加关注相关性更高的证据文档。模型仅凭随机初始化，便能联合学习检索和重建。在零样本设置下，该模型能够进行文档翻译，在*WMT2019* 德英翻译基准上取得最高 35.8 的 BLEU 分数；还能够执行抽象式摘要、问答和改述。在多种语言的多项任务上进行微调后，模型性能进一步提升，表明 MARGE 是一种具有广泛适用性的预训练方法。

XLNG [32] 使用MLM和翻译语言模型损失（表3.1）同时对同一个序列到序列模型进行预训练。预训练目标为不同语言在共同空间中生成嵌入，从而实现零样本跨语言

转移。在微调阶段，使用单语数据在自然语言生成任务上训练预训练模型。这样，在单一语言上训练的模型就能直接解决其他语言中的相应任务。对于零样本跨语言问题生成和抽象式摘要生成，该模型的性能优于基于机器翻译的方法。此外，这种方法还通过利用资源丰富语言的数据，提升了训练数据较少的语言的性能。

3.3.3 自回归语言模型

GPT-3 等生成模型在大量文档集合上进行训练，这些文档通常包含不同语言的文本。通过这些训练数据，模型还获得了关于这些语言的知识，并生成联合的上下文意义表示。如第3.6.3节所述，只要提供适当的提示和一些示例（少样本学习），它就能够在不同语言之间进行翻译。例如，在 WMT2016 英语→德语任务中，GPT-3 的少样本 BLEU 得分达到 29.7，而监督式最先进水平为 41.2；在德语→英语方向上，GPT-3 以 40.6 的 BLEU 得分超过了当前 40.2 的最先进水平 [25]。

Winata 等人 [231]详细评估了 GPT-2、GPT_{NEO} 和 T5 的多语言能力，它们分别具有 1.6B、6B 和 3B 个参数。这些模型能够利用来自英语的上下文来预测非英语语言中的回答。作者发现，在一组多语言基准测试中，最大的模型 GPT_{NEO} 始终表现最佳。性能取决于语言对。例如，与另外两种目标语言（德语和法语）相比，这些模型在英语→西班牙语上的性能更高。对于*MultiNLU*基准 [187] ，在目标语言上进行完整训练的最先进模型错误率为 12.1%，并没有比 GPT_{NEO}少样本提示的 17.3% 错误率低太多。

3.3.4 总结

机器翻译是自然语言处理中应用最广泛的领域之一。不同语言在结构和词汇上存在差异，这使得翻译变得困难。对多种语言进行联合处理时，必须考虑这些差异。

使用多种语言的文档训练 BERT 时，即使无法获取平行文本，它也能够在不同语言之间转移知识，例如完成语言推理任务。XLM 通过使用翻译语言建模损失来提升知识转移能力，即利用翻译后的句子重建掩码词元。XLM 有多个改进版本，能够提高跨语言推理的准确率。

T5 等编码器-解码器模型可以推广到多种语言，并生成强大的多语言嵌入。mT5 可以通过前缀进行控制，并

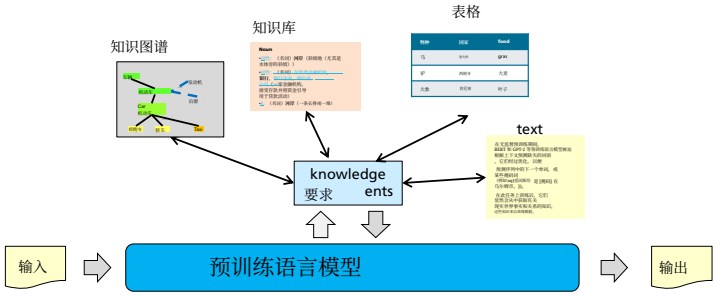


图 3.12 预训练语言模型接收输入文本，并从不同来源收集额外知识。这些知识可以预先添加，也可以根据需要检索。随后，利用额外知识生成输出。

解决翻译、摘要生成和语言推理等各种任务。mT6 和 Switch 是 mT5 的更高效变体。mBART 通过恢复不同语言中的损坏文本进行预训练。它甚至可用于无监督机器翻译。XNLG 在多语言空间中生成联合嵌入，而 MARGE 则利用检索到的背景文档来重建目标文档。这两种模型都能执行多项任务，例如抽象式摘要、问答和改述。不过请注意，翻译单一语言对时会使用专用模型（第 6.3.1 节）。

GPT-3 等自回归语言模型在海量语料库上进行训练，其中也包含多语言文档。因此，也可以通过少样本学习指导这些模型执行翻译或问答等多语言任务。然而，其性能通常不如专门的微调模型。

3.4 预训练语言模型的额外知识

在无监督预训练过程中，BERT 和 GPT-2 等预训练语言模型需要根据上下文预测缺失的词语。它们经过优化，能够预测序列中的下一个词，或某些掩码词（例如 “爱因斯坦曾 [掩码] 在乌尔姆市。”）。通过在这项任务上训练，它们显然从训练数据中获取了有关现实世界事实和关系的知识。预训练语言模型在无监督训练后，重现事实和关系的表现出奇地好。在第 4.2 节中，我们将讨论标准预训练语言模型涵盖了哪些知识。然而，由于参数数量仍然有限，预训练语言模型只能记住训练数据所含知识的一小部分。此外，它们无法掌握训练之后发生的事件。

本节介绍了扩展预训练语言模型（PLM）中事实知识的方法，这些方法既可在训练期间使用，也可在模型实际使用过程中动态应用（图 3.12）。一个知识

库 （知识库）描述了关于世界的知识，例如实体及其关系。我们概述了多种不同的方法，可将知识库或其他知识来源（如文本集合）中的信息整合到预训练语言模型中（表 3.9）：

知识库嵌入：有一些技术可以通过嵌入来表示知识库中的实体和关系。许多方法尝试将这些嵌入与预训练语言模型创建的词元嵌入相结合。这样，知识库中的信息便可以注入预训练语言模型，并用于下游任务。

表格的文本编码：表格中通常包含额外知识。这些表格中的条目可以采用特殊的文本格式进行编码。预训练语言模型可以使用这些文本进行训练，以获取行和列中的知识，这与学习两种语言中词语之间的关系类似。知识库关系的文本编码：使用知识库信息的另一种方法，是先识别文本中的实体或概念。然后，可以从知识库中提取这些实体和概念所涉及的关系，并将其作为文本或以其他合适的形式纳入训练过程。

添加检索到的事实：当预训练语言模型需要回答问题或生成文本时，可以围绕主题构造查询，并从知识库或互联网检索相应的文本内容。
Transformer可以获取这些文本信息，并据此改进输出。这样，模型就能即时使用全面且最新的信息。
增强逻辑一致性：PLM 有时无法生成逻辑一致的内容。通过额外的微调任务，可以训练模型遵循逻辑一致性。

Dash 等人 [45] 和 Yu 等人 [243]综述了将领域知识融入深度神经网络的方法。

3.4.1 利用知识库嵌入

通常，知识库是图结构，其中节点对应实体，边表示连接实体的关系。近年来，许多大规模知识库相继发布，其中包含数百万个实体，例如*WordNet* [137], *YAGO* [200], *Freebase* [18], *DBpedia* [15],和*DiffBot*[77]。图 3.13 展示了 WordNet 层级结构的一个小子集。在大多数情况下，知识库可以用三元组（h，r，t）来描述，其中h 和 t 是集合 E中的实体，而 r 是这些实体之间成立的一种关系。为了评估知识库的语义内容，有人提出将其中的实体和关系编码为低维空间中的嵌入，从而能够确定实体和关系的相似度 [43]。随后，这些嵌入可用于消歧实体（实体链接，见第 5.3.3 节），或预测新的关系（见第 5.4 节）。

表 3.9 整合额外知识的模型（参见 [166, p. 10]）。基准测试：GLUE 自然语言理解，第 4.1.1 节；TACRED 关系抽取，第 5.4 节。2 [199], TriviaQA 问答，第 6.2 节。1 [99], 英语全词词义消歧 [14], Nat. Quest 问答 [109] 第 6.1.2 节

模型	训练任务	微调	额外	基准测试
在预训练语言模型中使用知识库嵌入				
ERNIE（清华大学） [255]	掩码语言模型+下一句预测 + 掩码命名实体	GLUE 等	知识库命名实体嵌入与词元嵌入	GLUE 79.6
KnowBERT [157]	掩码语言模型+下一句预测 +实体链接 GLUE 等		转换词元嵌入 ↔知识库命名实体嵌入	
KEPLER [224]	MLM+KE	GLUE 等	将词元嵌入与命名实体结合嵌入；使用 TransE 损失	TACRED 71.5 F1
使用来自知识库的文本信息				
K-Adapter [222]	掩码语言模型 +关系抽取	-	向 RoBERTa 添加并行适配器网络	TACRED 72.0 F1
WKLM [234]	掩码语言模型+实体替换检测	-	检测文本中被替换的命名实体	TriviaQA 63.1 F1
CoLAKE [202]	MLM	-	根据文本关系三元组创建图和词元	GLUE 86.3
LUKE [234]	掩码语言模型+实体替换检测	-	针对文本和包含的实体	TACRED 72.7% F1值
EWISER [14]	MLM	词义分类	包括 WordNet 上位语义图	全部英语单词词义消歧 80.1% F1值
使用从文本集合中检索出的文本段落				
芬兰语 [91]	掩码语言模型、序列到序列	QA	使用 BERT 对查询和知识库进行编码；合并查询和检索文档与序列到序列	Nat. Quest. 51.4% 准确率
Retro [21]	LM		周期性语言生成检索	Nat. Quest. 45.5% 准确率

对于 Word2Vec [135]生成的词语嵌入emb(word)，研究发现，实体之间的关系通常表示为词语嵌入空间中的实体嵌入向量差（第 1.5 节）。例如国家与其首都之间的关系，近似而言emb(德国) $-$ emb(柏林) $\approx$ emb(法国) $-$ emb(巴黎)。

TransE 模型 [20] 建立在这一模式之上。TransE 调整嵌入，使得只要 (h, r, t) 成立，且 emb(h) 和 emb(t) 分别是 h 和 t 的嵌入，那么对于某个向量 emb(r)，等式 emb(h) +emb(r) $\approx$ emb(t) 应当近似成立，其中该向量被视为关系 r的嵌入。因此，对于集合 S中的所有正确三元组 (h, r, t)，TransE 损失$f_r(\mathrm{h}, t) = \|\mathrm{emb(h)} + \mathrm{emb}(r) - \mathrm{emb}(t)\|_2^2$ 应当变为 0。TransE 模型使用铰链损失来近似这一目标，通过调整嵌入，使得对于正确关系三元组，$f_r(\mathrm{h}, t)$ 低于随机选取的不正确三元组 $(\tilde{\mathrm{h}}, r, \tilde{t})$ 的 $f_r(\tilde{\mathrm{h}}, \tilde{t})$。模型和嵌入使用来自 WordNet 和 Freebase 的关系进行训练。

综述中介绍了多种更复杂的模型，用于编码知识库中的关系 [43, 94]。*TransH* 克服了 TransE 无法对复杂关系建模的缺陷，而*TransD* 则通过为注意力头和尾巴提出两种不同的映射矩阵，旨在减少参数。不过，这些替代方法很少用于上下文嵌入。另一种知识库表示方法是张量分解——

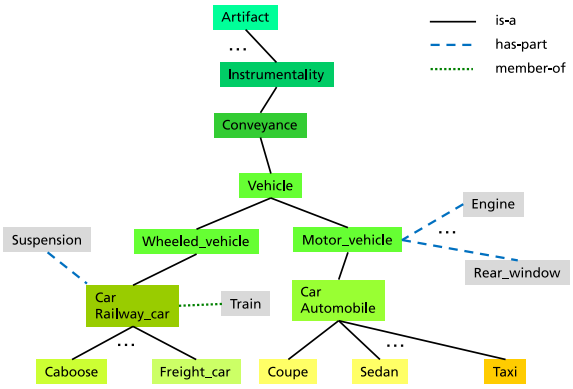


图 3.13 WordNet 知识库的一小部分，描述英语单词之间的关系。它包含意义大致相同的单词的同义词集，这些同义词集通过上位词（是某种）、部分词（具有某部分）和成员关系 [137]相互关联。

分解 [144, 145]。不过，这种方法并非基于词嵌入，因此主要用于知识库补全，而非增强预训练语言模型。

在本节的其余部分，我们将介绍一些方法，这些方法融合了通常由 TransE 计算得到的知识库嵌入，以及由语言模型生成的词元嵌入。一个难点是建立词元嵌入与实体之间的关系，因为实体通常包含多个词元。

KEPLER [224] 由一个类BERT语言模型构成，该模型通过掩码语言模型目标生成词元嵌入。此外，它还利用输入文本开头的特殊词元 “<*S*>”，根据知识库中的描述性文本计算实体的嵌入。该词元经过训练后，可以生成关系中命名实体参数的嵌入，例如图 3.14 中输入 “<*S*> *Johannes Kepler*” 的情况。通过这种方式，关系的参数 h 和 t 都会被嵌入。关系 r 的嵌入可以是待训练的参数，也可以根据将关系语言化的文本确定。这些嵌入会被输入 TransE 损失，并作为除掩码语言模型之外的额外训练准则使用（图 3.14）。在多个语言理解任务中，该方法都取得了良好的结果。在关系抽取基准 *TACRED* [254] 上，该方法达到了 71.5% 的 F1值。

KnowBERT [157]显式地对输入文本中的实体跨度进行建模，并使用实体链接器从知识库中检索预计算的实体嵌入，以形成知识增强实体跨度表示。知识库嵌入使用类似 TransE 的损失函数进行预计算。投影映射用于在语言模型嵌入与知识库嵌入之间进行相互转换。对匹配度最高的知识库嵌入中的信息取均值，并将其重新转换，以增强语言模型嵌入。这些计算构成了 BERT 的一个额外层。维基百科和 WordNet 被用作知识库。为了测试 KnowBERT 从知识库中检索事实的能力，研究者构造了一个关系，并对该关系的一个论元进行掩码。KnowBERT 达到平均倒数排名 (*MRR*) 0.31，这表明正确实体的平均排名为第 3 位；而对于 BERT，正确实体排在第 9 位。因此，该模型生成的答案优于 BERT，但只能近似复现知识库中的关系。不过，它通常能改善下游任务的表现。

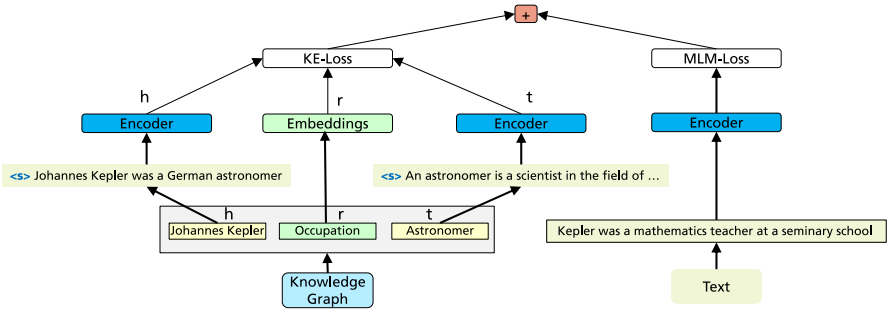


图 3.14 KEPLER [224] 通过掩码语言模型损失训练传统的类 BERT 模型。对于包含文本条目的知识库，它使用特殊的 *<S>* 词元生成实体嵌入，并通过 TransE 损失对关系进行编码。训练期间会同时加入这两个损失函数。

ERNIE-THU [255] 以类似的方式，将知识库中的命名实体与文档中的命名实体关联起来，并在这两个空间之间转换嵌入。*E-BERT* [162] 在理念上类似于 KnowBERT，但不需要对 BERT 编码器进行昂贵的额外预训练。*Facts as Experts* [213] 同样使用嵌入来链接事实信息与实体，从而能够将新信息注入模型。

总之，本节介绍的方法将关系嵌入所表达的领域知识直接融入预训练语言模型的词元嵌入中。然而，这些方法存在一些缺点。知识库实体嵌入使用某些知识嵌入模型（例如 TransE [20]）单独进行预训练，并在预训练语言模型的训练期间保持固定。因此，知识库嵌入与词元嵌入并非同时学习。此外，知识库实体嵌入往往无法充分捕捉知识库中实体丰富的上下文信息和关系信息。进一步说，这些嵌入是静态的，不随上下文而变化。此外，它们在很大程度上依赖链接算法的性能和图嵌入的可靠性。这意味着，总体而言，其他方法的表现通常更好，例如在关系抽取方面（第 5.4 节）。

3.4.2 用于图学习的预训练语言模型

物体与概念之间的关系可以连接成图，并为众多项目之间的关联性提供统一表示。利用图的结构，可以预测节点的许多性质。近年来，人们付出了大量努力来设计能够捕捉图的组成并预测其各部分的模型，例如 *node2vec* [67] 或 图卷积网络 [107]。然而，此类深度模型获得的节点表示往往会过度平滑，也变得非常模糊。预训练语言模型有望通过以下方式改进表示

对长距离进行自注意力。Xia 等人 [233] 对用于图的预训练语言模型进行了综述。节点和边通过不同的特征与位置嵌入进行表征，并使用不同类型的预训练语言模型处理。突出的应用包括推荐系统 利用用户-产品图，以及药物发现 评估分子结构。

Graph-BERT [250] 基于从大型图中抽取的样本节点及其上下文进行训练。这些样本根据 PageRank 算法 [24] 计算的接近度抽取，不包含直接链接信息。节点由特征嵌入、基于 PageRank 信息的嵌入以及基于跳数的距离嵌入进行表征。这些嵌入经过汇总后，构成 BERT模型的输入。该模型经过预训练，用于重建掩码节点的信息，并通过评估两个节点之间的余弦相似度来预测它们之间的关系。该模型针对节点分类和图聚类进行微调。Graph-BERT 在三个图基准测试中的节点分类任务上取得了第二高的准确率 [128, 第 16]页。

GPT-GNN [87]提出了一种自回归预训练语言模型，用于对给定图执行迭代重建。该方法假设边和节点具有随机顺序。给定截至特定位置的边和节点后，它会预测下一个节点或边的性质。GPT-GNN 每次生成一个掩码节点及其边，并通过最大化当前迭代中生成的节点和边的似然来优化参数化模型。随后，它不断生成节点和边，直到生成所有掩码节点。该模型在一个包含 1.78 亿篇科学论文及其特征、发表场所和作者的图上进行训练，同时还在一个包含 8300 万条亚马逊评论、用户和产品的图上进行训练。在这两个基准测试中，该模型都取得了最高的准确率。

MPG [120] 由一个对节点和边特征进行编码的 BERT模型组成。作为一项预训练任务，该模型需要学习两个被分成两半的图是否确实属于同一整体，还是这两半只是随机配对。该模型用于分子建模，并在一系列共 14 项基准测试中取得最先进水平的结果，尤其是在药物发现领域。

GraphFormers [238] 将图结构与词语序列联合建模。图中的每个节点都包含文本。中心节点及其邻居会被分词为词元序列。该模型包含专门的 Transformer 层，用于计算文本词元的嵌入，并通过聚合相应的文本嵌入来推导节点嵌入。模型通过预测两个节点是否相连的任务进行预训练。GraphFormers 在三个基准任务上进行了测试，例如，包含科学论文的图，其中的论文由标题及其引文图表征。在链接预测任务中，该模型始终优于此前的所有方法。

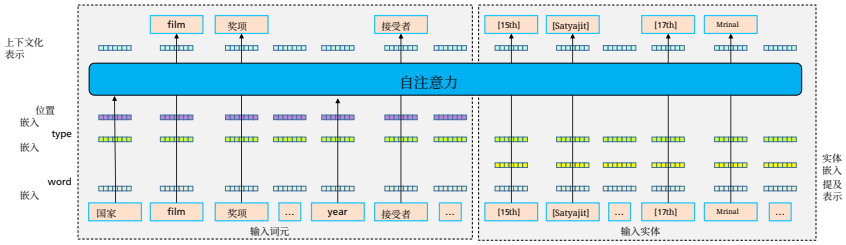


图 3.15 使用 TURL [47]学习表格关系。左侧训练表格标题和列标题。右侧处理行标记以及输入实体（特定行中的单元格）。

3.4.3 表格的文本编码

如今，表格数据可能占所有商业和行政数据的大多数。例如，零售交易、官方统计数据、工业应用中的处理数据等。de Alwis 等人 [46] 对 Web 上的表格解读进行了综述。以往的工作通常依赖人工选择的特征，无法处理网页表格中灵活多样的模式，而且在不同任务间的泛化能力较差。

TURL [47] 通过表格标题 C（一段简短文本，可由章节标题增强）、描述表结构的列标题 h_i（词元序列）$H = \{h_1, \ldots, h_m\}$以及单元格值来表征关系表，其中每个单元格都可能表示一个实体，例如一个人。同一行中的单元格共享某种关系，同一列中的单元格共享另一种关系。这需要通过可见性矩阵实现具备结构感知能力的注意力机制，将注意力限制在特定的列和行中。

TURL 在由表格标题和表头组成的大型非结构化数据集上，根据掩码语言模型损失进行预训练。随后，它可以学习同一行或同一列中实体之间的关系。表格中的实体会被掩码，模型的任务是根据表格上下文和可见性矩阵预测这些实体。借助这一目标，TURL 可以从表格中学习事实关系，并将其编码为实体嵌入（图 3.15）。

该模型使用从维基百科提取的 57 万个表格进行训练。所有至少包含一个已链接单元格的列都会被标记为实体列。经过微调后，该模型能够以 54.8% 的精确率预测测试集中的掩码表格单元格内容，优于其他竞争方法。消融研究表明，可见性注意力矩阵对于实现高性能至关重要。

TaBERT [241] 旨在同时纳入自然语言文本和结构化表格数据。TaBERT 使用来自英文维基百科和 WDC WebTable语料库 [115]的 2660 万个表格及其周边文本进行训练。每个表格单元格都表示为（列标题、列值类型、值）。随后，表格行被编码



Year	发表场所	位置	赛事
2005	埃尔福特	1st	欧洲U23锦标赛
2006	莫斯科	2nd	世界室内锦标赛

[CLS] 上下文... [SEP] 年份 | 实数 | 2005 [SEP] 发表场所 | 文本 | 埃尔福特 [SEP] 位置 | 文本 | 1名次 [SEP] ...

图 3.16 TaBERT 将表格的各行编码为特定格式的文本。“上下文”包含对应的文本。每个表格单元格都表示为（列标题、列值类型、值）。此处，第一行表格由以 [CLS]开头的行编码。

为文本，如图 3.16 所示。在预训练期间，随机选取表格中 20% 的列，模型需要预测被掩码的列名和类型。此外，还会按照一种特殊方案重建单元格值。模型在*WikiTableQuestions* 基准 [155], 上进行微调，该基准包含一些问题，需要对给定表格中的一系列条目进行组合式多跳推理。为减少工作量，只对包含查询词元的表格行进行编码。TaBERT 能够将该基准上的最先进的准确率提升至 51.8%。作者表明，其表格单元格编码比其他方案更有效。**RPT** [205]提出了类似的表格编码方案。**BRIDGE** [124]是一个语义解析系统，可将文本和表格中的信息转换为 SQL 查询，从数据库中提取信息。

Tapas [81] 是针对表格处理优化的 BERT 变体。表格按行展平、分词，并添加位置嵌入。还会添加以下嵌入：行 ID 嵌入、列 ID 嵌入，以及用于指示排序序列中排名的排名嵌入，例如数字的排名。该模型使用来自英文维基百科的 620 万组表格-文本对进行预训练，任务是恢复表格和文本中被掩码替换的词语。该模型的表现相当准确（在测试集上的准确率为 71.4%）。

在微调过程中，模型学会从表格中回答问题，例如对于包含摔跤比赛结果的表格，“哪位摔跤手的冠军统治次数最多？”[*CLS*]和查询会被添加到扁平化的表格前，并通过额外的片段嵌入来区分这两部分。模型有两种输出类型：1）为每个表格单元格提供一个得分，表示该单元格属于回答的概率；2）提供结果类型（无、计数、求和、平均值）的概率，以便 [*CLS*] 生成最终回答。两者结合起来，可以确定应对哪些表格单元格执行哪种操作，从而生成最终回答。在多个基准测试中，Tapas 达到了最先进水平的结果，例如在 SQA 基准测试中，得分从 55.1% 提高到了 67.2% [90]。源代码和预训练模型可在 Hugging Face 上获取。

结果表明，上述模型能够从表格中提取信息，并回答有关表格内容的问题。这使得利用大量信息来源成为可能，因为表格在文本文件和网页中无处不在。原则上，这种方法也可以用于让大型基础模型将表格信息纳入其生成的文本中。

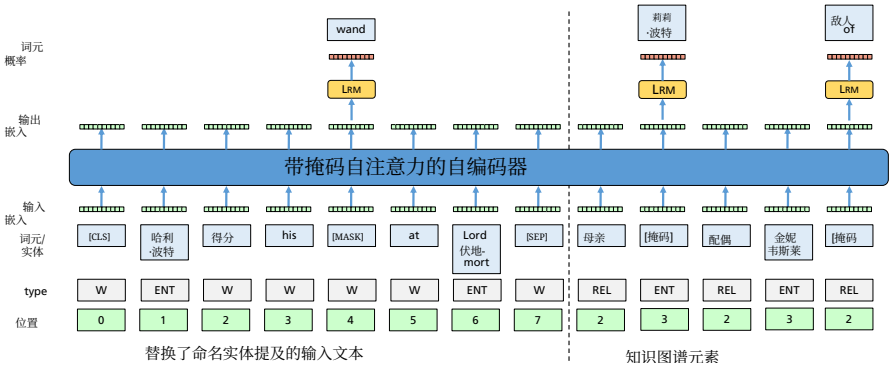


图 3.17 CoLAKE [202] 识别实体，并使用特定嵌入对其进行编码。类型嵌入用于区分词语、实体和关系。输入嵌入是词元/实体嵌入、位置嵌入和类型嵌入之和。对于输入文本中的所有实体，都会从知识库中提取关系，并附加在“[*SEP*]”之后，例如母亲（哈利·波特，莉莉·波特）。掩码机制确保关系元素只能关注输入文本中与之对应的元素。在预训练期间，模型必须预测被掩码的词元和实体。

TableGPT [63] 使用 GPT-2 语言模型从表格生成文本。它通过表格结构重构和内容匹配这两项辅助任务增强 GPT-2 的表格到文本生成能力，从而提高文本忠实度。

3.4.4 知识库关系的文本编码

许多方案尝试用文本来表达知识库关系。这样，知识库关系便可直接纳入语言模型的训练文本中。

WKLM [234] 会随机将原始文档中的一部分实体提及替换为其他同类型实体的姓名。模型经过训练，能够从正确的实体提及和随机选取的实体提及中辨别出前者。此外，模型还必须预测掩码词元。实体类型取自维基数据 [214]。通过这种方式，模型能够更好地从自然语言中捕捉实体信息，并在实体相关的自然语言处理任务中取得更好的结果。WKLM 对关系论元的预测能力远胜于 BERT。在问答任务中（SQuAD 和开放域，见第 6.2 节），该模型也能达到最先进水平。类似的方法 [191, 203, 234] 提出了实体和短语掩码及替换方案。

CoLAKE [202] 从大规模知识库中提取实体的知识上下文。该模型通过知识库中的实体链接器，将实体提及链接到知识库中的相应实体。随后，提及节点被其链接的实体替换。CoLAKE 模型以 $\text{RoBERTa}_{\text{BASE}}$ 模型进行初始化。它在维基百科上进行训练，其中包含 300 万个实体嵌入和 822 个关系嵌入，并与

知识库*Wikidata5M* KB [224] 中的 2600 万个训练样本对齐。示例输入 "[*CLS*]哈利·波特用魔杖指着伏地魔 [*SEP*]" 如图 3.17 所示。输入的类型（单词、实体、关系）被编码为类型嵌入，并添加到词元嵌入和位置嵌入中。为了引入知识库中的关系，例如"*(*哈利·波特，母亲，莉莉·波特*)*"，我们引入关系节点"母亲"和实体节点"莉莉·波特"，并为它们设置位置嵌入 2 和 3，因为第一个关系论元"哈利·波特"位于位置 1。文本输入之间会计算自注意力。对于关系元素，存在一种限制自注意力的掩码机制，例如在本示例中限制于"*(*哈利·波特，母亲*)*"以及"*(*母亲，莉莉·波特*)*"这两对。

在预训练期间，约 15% 的输入元素（词语、实体、关系）会被掩码，并由模型进行预测。由于实体节点会同时出现在输入文本和知识库中，这有助于对齐语言表示与关系表示。对关系节点进行掩码，有助于 CoLAKE 学习关系的上下文化表示。在 GLUE 的语言理解任务中，CoLAKE 模型取得了与 RoBERTa 相近的平均分 86.3。另一种任务是利用描述关系的句子补全关系三元组（h, r, t）。结果表明，CoLAKE 远胜于其竞争对手，例如，在 72.1% 的案例中，模型能够根据两个实体推断出正确的关系。

LUKE [237]将给定文本中的词语和实体视为独立的词元，并输出二者的上下文化表示。该模型基于 BERT，并在大型维基百科实体标注语料库上训练，以预测随机掩码的词语和实体。它包含一种实体感知自注意力机制，是对 BERT 自注意力机制的扩展。该机制会考虑用于指示词元表示文本还是实体的嵌入。LUKE 在关系分类、实体类型判定和命名实体识别方面取得了最先进水平的结果。**K-adapter** [222] 是一种相关方法，它使用 RoBERTa（第 3.1.1 节）作为固定的背景模型，并构建多个相互独立的"适配器"，以引入来自不同知识库的知识。

EWISER [14] 同样针对词义消歧（WSD）。它从BERT嵌入开始，为WordNet同义词集（意义相近的词语集合）计算得分。利用同义词集图的相互依赖关系，该方法计算单词属于某个同义词集的最终得分。在多个词义消歧基准测试中，它达到了新的最先进水平（第5.2节）。

PET（模式利用训练）[184]作为一种替代方法，仅使用少量带标签示例来构建额外的训练集。以 Yelp 数据集中的餐厅五星级评分为例 [185]。作者在评论中添加文本来表达评分，例如 "总的来说，这很棒"。通过这种方法，作者将 Yelp 数据集转换为一个预测掩码词的任务，例如 "总的来说，这" [*MASK*]"。不过，他们只为少量示例提供了用语言表达的标签。随后，他们为未标注示例预测最佳类别，并使用预测类别以及语言建模损失来训练模型，以避免灾难性遗忘。这一过程可以进行多次迭代。尽管只使用了少量标签，但该模型在 Yelp 上的表现优于标准监督方法。SuperGLUE 基准涵盖八项具有挑战性的自然语言处理任务。仅使用 32 个带标签示例，按照上述方法训练的 PET 方法

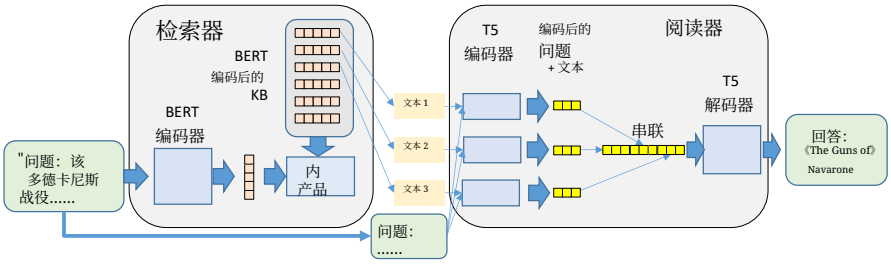


图 3.18 检索增强语言模型 [91] 将查询和知识库段落编码为嵌入，并使用预训练的检索器查找与查询对应的段落。阅读器是一个序列到序列模型（T5），它结合查询和段落来生成回答。该模型设置使用不同的基准数据集进行微调。

方案的平均得分（75.4%）高于使用相同数量少样本示例的 GPT-3（71.8%）。这表明，使用较小的模型（223M）和少量带标签示例也能取得良好结果。需要注意的是，使用 T5 和 Meena 在 SuperGLUE 上微调得到的最先进水平为 90.4%。

TeKGen [1] 是一种数据到文本的序列到序列模型，用于将完整的知识库用语言表达出来。该模型应用于英语*Wikidata* 知识库 [214]，其中包含 ≈6M 个实体和约 1500 种关系。模型以大型训练语料库为基础，其中包含通过启发式方法对齐的维基百科文本和维基数据三元组。共享同一实体主语 的关系会转换为 T5 Transformer（第 3.1.3 节）的输入：主语 关系$_1$ 宾语$_1$, ……，关系$_n$ 宾语$_n$。例如，“杀死一只知更鸟，作者：哈珀·李，发布日期：*1960* 年 *7* 月 *11* 日”会被转换为“《杀死一只知更鸟》是哈珀·李创作的一部小说，于 *1960* 年出版。”T5 模型经过微调并接受附加检查，以生成优质的文本化表述。由此得到的三元组文本化数据集被用于问答任务，并使*Natural Questions* 基准测试 [109]（第 6.1.2 节）上的准确率从 38.8% 提高到 41.5%。**KGPT** [30] 以类似方式将结构化知识转换为序列化文本，并让模型学习知识-文本对齐。

总之，这些方法会将知识库关系转换为文本，例如转换为表达关系的完整句子，或转换为串联的三元组（例如， [头部文本、关系文本、尾部文本]），再输入语言模型进行训练或微调。这些文本会被转换为上下文嵌入，并训练模型检测其中隐含的关系。其缺点是，专注于知识库补全往往会使模型过度适应这一特定任务，代价是泛化能力下降。

3.4.5 通过检索文本增强预训练语言模型

开放域问答系统的任务是回答不受特定领域限制的问题 [27]。请看以下来自 TriviaQA 基准测试的示例 [99]。“问题：第二次世界大战期间，多德卡尼斯战役是盟军试图夺取爱琴海诸岛的一次行动，这场战役启发了哪部广受好评的 1961 年突击队电影？”“回答：《纳瓦隆大炮》”。即使模型在训练前接触过相关数据，也很难指望它能从参数中存储的知识里复现如此具体的响应。因此，理想的做法是让系统能够通过检索器从大型文本库中收集相关文档，以获取更多证据。随后，系统需要将检索到的信息与问题对齐，并由另一个预训练语言模型（即阅读器）生成答案。网络搜索可以采用新的技术来实现这一方法。这些技术基于比较词语或由多个句子组成的段落的嵌入。其应用十分广泛，包括问答、摘要生成和对话系统。第 6.1 节将对此进行更详细的讨论。Zhu 等人 [259] 和 Yu 等人 [244] 提供了近期综述。

DPR（密集段落检索器）[103] 使用预训练语言模型对知识库段落 d_i（例如来自维基百科的段落）进行编码，生成嵌入 $emb(d_i)$。具体而言，可以通过微调 BERT模型，将词元 [*CLS*]嵌入来对段落进行编码。这些嵌入可以存储在索引中，以便快速获取。随后，DPR 检索器 通过另一个 BERT模型处理查询序列 x，并生成查询嵌入 $emb(x)$。通过近邻搜索，可以检索出若干与 $emb(x)$ $emb(\text{c})$ 内积最大的段落。BERT 编码器可以共同进行训练，通过问题-回答对形式的弱监督来生成适当的嵌入（参见第 6.1.5 节）。例如，当查询为 “指环王中的坏人是谁”时，算法可以检索出“*Sala Baker* 因在《指环王》三部曲中饰演反派索伦而广为人知”，因为“坏人” 和“反派”具有相似的嵌入。因此，DPR 能够找到意义相似但由不同词语表达的段落。例如，Karpukhin 等人 [103], 表明，仅使用 1,000 个训练样本，稠密检索器的效果就已经优于经典的关键词搜索。使用 4 万个训练样本时，排名前 20 的检索段落约有 79% 的概率包含正确回答，而经典检索的这一数值仅为 59%。第 6.1.5 节对此进行了深入讨论。

DPR 阅读器是另一个 BERT 模型。与 BERT 的文本对分类类似，它经过微调，可为每个检索到的段落预测一个概率，表示该段落包含正确答案的可能性。此外，它还通过片段预测选取一段词元，这段内容很可能就是答案。在此示例中，它选取了“*Sala Baker*” 作为回答。这两个组件共同构成了检索器-阅读器架构，这种架构近来颇受欢迎。该方法可以轻松应用于包含数十亿个段落的知识库 [103, 201]。在*Natural Questions* [109] 上，它在测试集上的准确率为 41.5%。

DensePhrases 是一个不同的系统，它为知识库中的长度不超过 20 个单词的短语创建嵌入，这些嵌入的计算不依赖于查询 [114]。对检索到的短语进行处理，便可直接得到回答，而无需大量计算

开销。通过仔细的工作流优化，作者以远低于密集段落检索系统的处理时间取得了接近最先进水平的结果，例如在 Natural Questions 测试集上的准确率达到 40.9%。

FiD（解码器中的融合） [91] 采用 DPR 作为检索器。在阅读器步骤中，它使用特殊词元*“question:”*、*“title:”*和*“context:”*。这些词元标记问题、检索到的段落标题和段落文本，并将它们串接起来形成输入。随后，这些 k 检索到的三元组逐一输入类似 T5 [170] （770M 参数）的 Transformer 编码器，由编码器分别处理每个三元组。只有在解码器中，段落才会被共同处理，并生成答案文本。这种方法大幅减少了计算开销。该 Transformer 在问答任务上进行微调。模型架构如图 3.18 所示。Raffel 等人 [170] 提供了证据，表明像 T5 这样的生成模型在 SQuAD 等问答任务中也具有竞争力；在这些任务中，答案是给定文档中的跨度。[173],

该系统在 Natural Questions 基准上的测试集精确匹配准确率为 51.4%，而 DPR 为 41.5%。*TriviaQA*基准 [99]包含一组问答题，其答案最初是从 Web 抓取的。在这一基准测试中，该模型取得了最先进水平的结果，精确匹配准确率达到 80.1% [211]。这优于其他规模大得多的模型，例如具有 1750亿参数的 GPT-3（71.2% EM），以及未使用检索、具有 11B 参数的 T5（60.5% EM）。事实证明，增加检索到的段落数量能够显著提升答案质量。

目前有多种新方法，可以利用外部知识库中的文本增强预训练语言模型。在第 6.1 节中，我们将介绍可供网络搜索引擎使用的不同检索预训练语言模型。在第 6.2 节中，我们将探讨问答系统；这类系统通常采用基于预训练语言模型的检索机制，并额外使用一个预训练语言模型来生成回答文本。它会结合查询、训练过程中获取的知识以及检索到的文档中的信息。

总之，将语言模型与检索相结合，是目前向预训练语言模型中融入额外信息最有效的方法。新增信息聚焦于当前查询，因此信息丰富度很高。借助新型开源近似最近邻索引结构，检索模型可以在不到一秒的时间内访问语义相关的段落。通过依赖嵌入，可以找到同义词和释义，并消除词语意义上的歧义。此外，底层知识库可以即时更新，以确保信息保持最新。

3.4.6 总结

文本训练数据所涵盖的知识可以通过多种方式加以利用，以提升预训练语言模型的性能。知识库中的实体和关系可以通过嵌入来表示，例如使用 TransE。不过，将这些嵌入用于预训练语言模型并不高效，也容易出错。一种更有前景的

替代方案是由专门的预训练语言模型直接使用表格内容或知识库关系，并通过特定的自注意力模式捕捉实体与表格单元格之间的关系。与图卷积神经网络类似，预训练语言模型也被直接用于获取图中节点之间的关系，方法是在类 BERT 模型中通过嵌入对链接的特征进行编码。沿着这一思路，GraphFormers 提出了一种将图中的关系知识转移到语言模型中的有前景的方法。

一种将表格和知识库纳入预训练语言模型的非常简单且高效的方法，是创作能够表达信息内容的文本。预训练语言模型可以将这类文本用作条件文本，或在训练期间使用。然而，纳入知识最有前景的方式是检索，因为大多数信息都以非结构化文本的形式存储在 Web 或数据库中。在此，检索器-阅读器架构已成为收集相关段落的有效方式。随后，预训练语言模型通过结合内部知识、起始文本和检索到的段落来生成新文本。

人们投入了大量精力来扩展输入序列的长度（第 3.2 节）。这主要通过稀疏注意力模式实现，使计算开销的增长从二次降至线性，其中 S4 是一种领先的方法。尽管如此，较长的输入序列在同一样本内部及其外部的上下文范围仍然有限。

相比之下，检索可以通过收集恰当的段落，在同一样本中覆盖无限范围的上下文，即使整个上下文并未同时参与注意力计算。此外，检索还可以获取海量文档集合中的相关信息。既可以使用高度成熟的传统关键词搜索引擎，也可以采用稠密检索：它利用索引上的近似最近邻搜索，对查询和段落的嵌入进行比较。事实证明，规模相对较小的基于检索的模型，其表现优于 GPT-3 等大型基础模型。例如，FiD 在 Natural Questions 基准上的精确匹配准确率达到 51.4%，而 GPT-3 为 29.9%。WebGPT 和 Retro 等近期模型广泛使用检索。

3.5 改变模型规模

模型的规模，尤其是其参数数量，会显著影响模型的性能、内存需求以及训练所需的计算资源。第一节将讨论参数更多的模型可能具有更好的性能，但这需要在训练和使用模型时投入更多的计算开销。另一种选择是混合专家模型，它们定义了多个并行的模型结构，并有选择地计算解决方案。第二节将对此进行介绍。

由于成功模型的初始版本往往规模极大，因此人们开发了多种模型压缩与加速技术。它们能够在准确率没有明显下降的情况下，减少内存需求和训练时间，并使模型得以部署在资源受限的计算设备上，例如手机

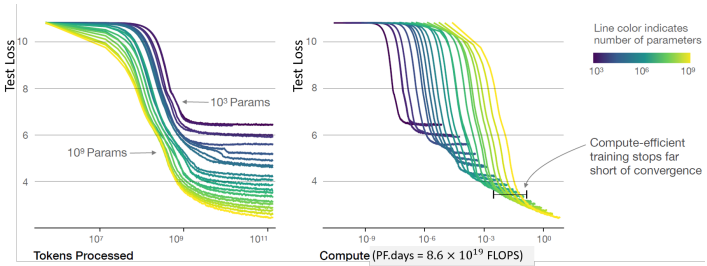


图 3.19 一系列采用不同模型规模的语言模型训练实验 [102]。左图显示，规模较大的模型只需更少的样本即可达到固定的测试损失。右图表明，模型规模应随着计算预算的增加而增大。经作者慷慨许可转载的图像 [102, 第 4] 页。

上。模型规模缩减主要有三种技术 [65]：参数压缩与缩减、低秩因子分解和知识蒸馏，后续章节将对此进行概述。

3.5.1 更大的模型通常性能更好

机器学习的一条经验法则是，应限制模型的参数数量以避免过拟合，也就是避免模型适应数据中的随机波动。然而事实证明，如果同时增加训练数据量和模型参数数量，这条法则并不适用于预训练语言模型。研究表明，规模更大的预训练语言模型在自然语言处理任务上性能更好，关于预训练语言模型的理论研究也印证了这一点 [19, 第 117] 页。增加参数数量带来的收益来自两个方面：训练和推理时进行更多计算，以及增强对训练数据的记忆。Kaplan 等人 [102] 通过实证详细研究了模型参数数量 R （不包括嵌入）、训练数据规模 N 与训练所用计算开销 C之间的关系。他们评估了大量模型，并得出以下结论：

- 模型的性能在很大程度上取决于规模量 R， N， C。宽度或深度等其他架构特征的影响则很小。
- 只要其他量不太小，性能就会与每个 R， N， C 呈平滑的幂律关系。例如，损失近似为 $L \approx (N/(5.4 * 10^{13}))^{-0}$。^{095}。

- 如果以相同速率增加 R 和 N，模型准确率就会稳定提升。如果其中一个因素保持不变，提升幅度就会降低。要获得最佳性能，当数据 N 增加 5 倍时，模型规模 R 应扩大 8 倍。

- 训练损失与计算开销之间存在可预测的依赖关系，因此可以进行外推。
- 预训练模型在不同训练任务上的微调性能，很大程度上取决于预训练验证集上的损失。因此，转移到不同分布会带来恒定的惩罚，但总体而言，性能会随着预训练集上的性能提高而提升。
- 大型模型比小型模型更能从数据中提取信息。它们只需更少的优化步骤和更少的数据点，就能达到相同的准确率。如果计算时间是固定的，但对模型规模或数据没有限制，就应该使用超大模型，并在收敛之前停止训练（图 3.19）。最佳批量大小取决于梯度噪声，这种噪声在训练过程中很容易测量 [132]，而且比之前假设的更大。

这些发现表明，更大型预训练语言模型取得成功是一种系统性特征。模型参数数量更多时，其样本效率远高于此前的认识；过去，过拟合曾是较小训练任务中的主要问题。这也解释了 T5、BigBird 和 GPT-3 等大型模型取得成功的原因。Hernandez 等人 [80]研究了从预训练到微调的经验缩放定律。图 3.20 绘制了过去二十年间一些深度学习模型的训练开销。

3.5.2 混合专家模型

如上所述，参数更多的模型通常能够实现更好的性能。在不增加训练开销的情况下增加参数数量的一种简单方法是采用**混合专家** 架构。Nowlan 等人 [147] 早在九十年代就提出了这一方法 [152]，它与决策树模型有着很强的相似性。该架构由一个门控模块和若干个架构相同但参数不同的专家模块组成。每个专家只专门处理数据的一个子集，而门控模块则将每个输入分配给适当的专家。具体而言，门控网络会计算专家上的概率分布，以表示每个专家处理传入输入的能力。如果实际上只使用少数几个专家模块，就可以降低计算开销。模型通过随机梯度下降进行训练；即使交换了某个专家，该方法也能在存在不连续性的情况下计算参数梯度。增加专家数量可以使计算成本保持不变，因为无论专家总数是多少，模型都会为每个输入选择相同数量的少数专家。该架构支持构建超大规模模型，对于专家分布在不同计算设备上的分布式系统尤其高效。

Clark 等人 [38] 分析了此类路由网络的理论性质，其中每个输入仅由网络中一部分参数对应的子网络处理。作者分析了三种不同的架构，并得出以下结果。

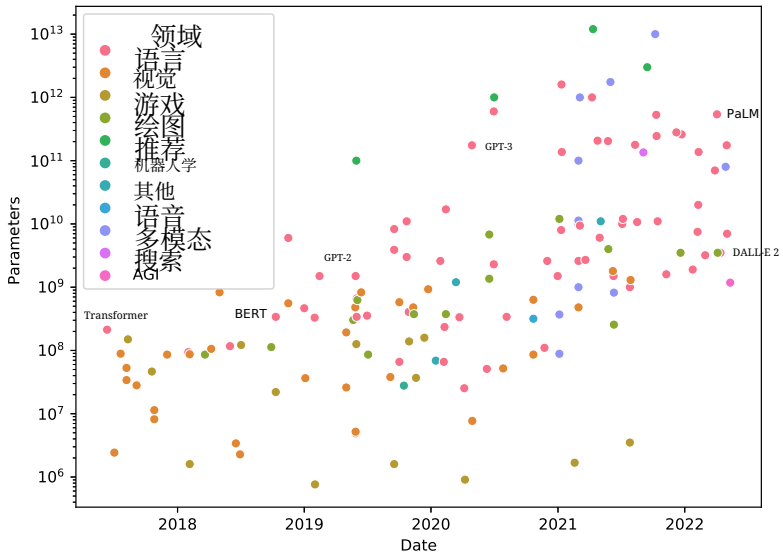


图 3.20 2017 [188]以来深度学习模型的参数数量。请注意，参数规模采用对数刻度。参数数量大致从 100M 增加到 1,000B。

- 在所有研究的规模和变体中，路由都能提升预训练语言模型的性能。
- 改进程度与专家数量 E 呈幂律关系，并且会随着模型规模 N的增大而减弱，还可以进一步推广到不同的路由架构。

该分析基于对多个规模层级的评估，其中包括拥有数百位专家和数千亿参数的模型。

GLaM [51] 是一种自回归混合专家（*MoE*）模型，参数量最高可达 1,200B。它将每隔一个编码器块（第 2.1.1 节）中的全连接层替换为 64 个参数各不相同的副本。对于每个嵌入，门控模块会从这 64 个全连接层中选择两个进行处理。该架构如图 3.21 所示。模型使用包含 1.6T 词元的海量文档和经过质量检查的网页进行训练。其参数量约为 GPT-3 的 7 倍，但训练开销仅为其三分之一。这样，模型便拥有更多参数，提升了表征能力。对于给定的输入词元，由于只使用两个专家模型，训练和应用的计算开销更低。在 29 项自然语言处理任务上，该模型的零样本和单样本性能优于 GPT-3。部分结果与其他模型的结果进行了比较

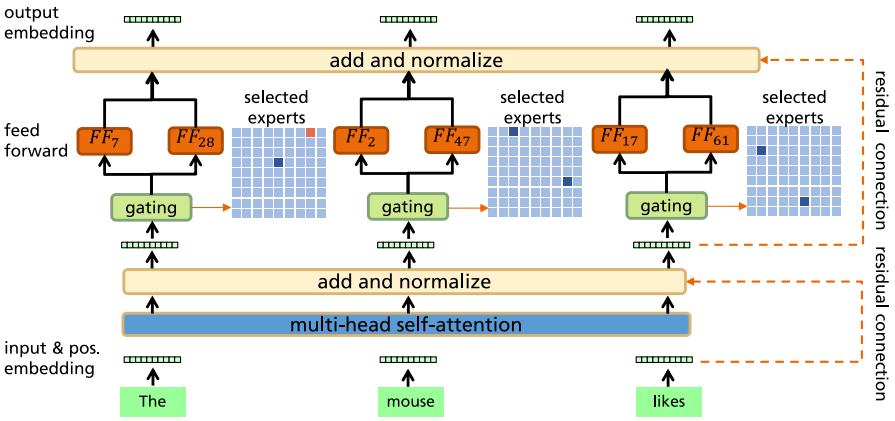


图 3.21 GLaM [51]的架构。对于每个输入词元，例如*"likes"*，门控模块会从可用的 64 个专家中动态选择最相关的两个专家。这由蓝色网格表示。随后，这两个专家的前馈模型输出的加权平均值会传递到下一个编码器块。对于其他输入，会选择不同的专家。每隔一个编码器块使用一个混合专家层。

见表 3.3 和 3.4。GLaM 的突出之处在于，其训练开销仅为 GPT-3 的三分之一，但在自然语言处理任务上的性能与 GPT-3 相当或更优。

悟道2.0 [175, 178, 257] 是一个近期的大型自回归语言模型，拥有 1,750B 个参数，是 GPT-3 的十倍大。它具有混合专家层，其中门控网络根据输入选择一个用于处理的子模块。悟道2.0 使用*FastMoE* 库 [74]，并采用 GLM 2.0架构（第 3.1.3 节），结合了 BERT、GPT 和编码器-解码器Transformer [175]的不同学习范式。

训练数据包括来自*Pile* 语料库 [61]的 1.2TB 中文文本、2.5TB 中文图形数据和 1.2TB 英文文本数据。使用*Cogview* 模型对图像进行联合处理（见第 7.2 节）。此外，悟道2.0能够即时学习、绘制图片和创作诗歌。这些能力与 GPT-3 有着显著区别。

已公布的性能主张令人印象深刻。在用于衡量世界知识的 LAMA 基准测试 [158] 中，它的得分高于 AutoPrompt [192]。在*SuperGLUE* 少样本自然语言理解任务 [219] 中，它达到了最先进水平，并超越了 GPT-3。在 Lambada 基准测试（见第 4.1.3 节）中，模型需要预测段落的最后一个单词，其结果优于 Microsoft Turing NLG。此外，它还在若干文本图形任务（见第 7.2.8 节）上刷新了最先进水平。

Switch [56] 是 Transformer 编码器-解码器 T5 的一个变体（第 3.1.3 节）。它采用混合专家架构，将每个编码器块中的全连接层替换为 k=128 个参数各不相同的副本。该架构还包含一个简单的线性门控网络，每个词元对应选择 128 个全连接层（即专家）中的一个。因此，参数数量大幅增加，同时

计算开销基本保持不变。对于这种架构，可以计算梯度，并使用若干特定策略和一个特殊版本的 TensorFlow 来优化模型。结果表明，Switch 达到与标准版 T5 相同的损失水平，而计算时间仅为其七分之一。在多项微调任务中，拥有 1,600B 个参数和 2,048 个专家的大型 Switch 模型，取得了优于 T5-large（第 3.1.3 节）的结果；后者有 13B 个参数，训练计算开销仅为前者的四分之一。

作为混合专家架构中门控网络的一种替代方案，可以使用哈希值激活网络的不同部分。**TokenSwitch** [177] 会为每个输入词元计算一个哈希值，并根据哈希值将每个词元生成的嵌入路由到不同的前馈网络。作者表明，他们的方法优于 Switch，并且在综合语言建模任务上表现良好。

ST-MoE-32B [261] 是一种混合专家模型，拥有 269B 个参数，其训练成本与 32B 稠密模型相当。作者修改了将词元嵌入分发给一位或两位专家的路由算法，并解决了稳定性问题。该模型类似于 T5-Large 编码器-解码器 [170]。ST-MoE-32B 有 32 位专家，专家层频率为 1/4，因此 T5 每四个前馈层中就有一个被混合专家层取代。作者使用*GEGLU* 激活函数，其中包含乘法元素 [142]

$$FFN_{GEGLU}(\boldsymbol{x}, W, V, \boldsymbol{b}, \boldsymbol{c}) = GELU(\boldsymbol{x}W + \boldsymbol{b}) \odot (\boldsymbol{x}V + \boldsymbol{c}). \tag{3.2}$$

作者比较了大量变体和超参数，以改进训练。

该模型在许多迁移学习基准测试中达到了最先进水平，例如在 SuperGLUE 上的平均准确率为 93.2%，超过了拥有 540B 参数的 PaLM 语言模型。在摘要生成（XSum [143]，ROUGE-2 为 27.1；CNN/Daily Mail [78]，ROUGE-2 为 21.7）、闭卷问答（WebQA [13] 47.4% 精确匹配，Natural Questions [109] 41.9% 精确匹配）以及针对常识推理构建的对抗性任务（WinoGrande [182] 96.6%，ANLI R 3 [146] 74.4%）中，也取得了其他最先进水平的结果。

3.5.3 参数压缩与缩减

模型量化 是一种参数缩减技术，其中参数以低精度存储，因此预训练语言模型中的计算也会更加不精确。传统模型通常使用 32 位或 16 位参数，而量化后的参数可以是 8 位，甚至是 1 位或 2 位。**Q-BERT** [190], 例如，将 Transformer模型量化为超低精度。这样可以将模型规模缩小 13 倍，同时仅损失 2.3% 的性能。作者避免了简单降低权重精度这一朴素方法，而是采用额外的训练步骤来调整量化后的权重，并允许对更“敏感”的参数使用更高精度。其他

作者提出删除值较小的参数 [64]。ALBERT [113] 在所有层中使用相同的权重，并实现了显著的参数缩减。然而，与 BERT 相比，ALBERT 的性能相同或更好。

另一种方法旨在减少参数数量，例如移除注意力头。研究表明，大多数注意力头只聚焦于几乎相同的位置关系，因此可以用固定注意力模式取代 [172]。事实证明，每个编码器单元只需 1–2 个注意力头，就能达到较高的性能，而原始模型中每个编码器单元有 16 个注意力头。Ganesh等人 [60] 对参数压缩技术进行了详细综述。

另一种减少模型参数的方法是模型剪枝，即移除预训练语言模型中无关的部分，从而在不损害性能的情况下减少内存占用并加快执行速度。例如，研究表明，可以移除 Transformer 的一些注意力头，而对准确率影响甚微 [256]。其他研究人员则对注意力层和线性层的权重进行剪枝，以减少参数数量而不降低准确率 [29, 64]。需要注意的是，模型剪枝并不总能带来加速，因为稀疏计算可能难以在图形处理器上并行化。

3.5.4 低秩分解

这种技术采用矩阵和张量分解来减少满秩参数矩阵的参数数量，此前已在第 3.2.2 节中讨论过如何借此扩展输入序列长度。示例包括 Performer [34] 和线性Transformer [105] （第 3.2.2 节）。作为另一种方法，ALBERT（第 3.1.1 节）将嵌入矩阵近似表示为两个较小矩阵的乘积。

3.5.5 知识蒸馏

在机器学习中，知识蒸馏方法 [82] 将知识从大型教师模型 迁移到较小的学生模型。大型模型通常可以成功训练，使其近似某种函数关系，而不必充分发挥其全部表征能力。为了降低应用过程中的高计算和内存需求，训练较小的模型来模仿大型模型，同时不牺牲准确率。

这种方法的优点是，可以训练学生模型来近似教师模型的内部激活。通常，教师模型生成的目标概率用于训练学生网络。对于输入 $\boldsymbol{x}$ ，教师模型的输出通常为 $z(\boldsymbol{x})$，通过缩放后的 softmax 可将其转换为概率

$$\boldsymbol{y}(x|\tau) = \frac{[\exp(z_1(\boldsymbol{x})/\tau), \ldots, \exp(z_k(\boldsymbol{x}))/\tau]}{\exp(z_1(\boldsymbol{x})/\tau) + \cdots + \exp(z_k(\boldsymbol{x})/\tau)}, \tag{3.3}$$

其中 $\boldsymbol{y}(x|\tau)$ 是一个概率向量，τ 是一个名为温度的参数，对于标准 softmax，该参数通常设为 1.0。学生模型通过最小化交叉熵来模仿教师模型生成的概率 $\hat{}\boldsymbol{y}(x|\tau)$

$$E(\boldsymbol{y}|\tau) = -\sum_{j=1}^{k} \hat{y}_j(x|\tau) \log y_j(x|\tau), \tag{3.4}$$

其中，$\boldsymbol{y}(x|\tau)$ 是学生模型的输出概率向量。如果有观测值，则教师模型 $y_j(x|\tau)$ 的概率可以替换为：观测类别为 1.0，其他类别为 0.0。训练期间可以调整温度。较高的温度可以避免极端概率值，并降低梯度。这可能使优化在初期更快收敛。

DistilBERT [183] 使用掩码语言模型交叉熵损失预测词元概率，并额外利用教师网络和学生网络的嵌入矩阵之间的余弦相似度来训练更小的 BERT 模型。它在预训练期间采用知识蒸馏，将 BERT 的体积缩小 40%，同时保留其 99% 的原有能力，并使推理速度提升 60%。*MobileBERT* [204] 基于一个特定的大型 BERT 模型，并转移有关多头注意力以及由此产生的嵌入的信息。实验表明，MobileBERT 的体积比 BERT 小 4.3 倍、速度快 5.5 倍，同时在知名基准测试中取得了具有竞争力的结果。

TinyBERT [97] 提出在预训练和微调期间对 BERT 模型进行蒸馏。该模型针对以下内容进行调整：1）选定层的嵌入输出；2）从选定的 Transformer 层中得到的隐藏状态和注意力矩阵；3）预测层的 Logit 输出。由于微调期间也会进行蒸馏，因此该模型能够更好地适配经过微调的 BERT。在多个基准测试中，TinyBERT 的表现与 $\text{BERT}_{\text{BASE}}$ 不相上下，并优于 DistilBERT。

请注意，上文讨论的知识蒸馏方法需要使用用于预训练教师模型的数据，而这类数据通常因数据版权问题而不会公开。目前尚未评估使用新数据进行蒸馏是否可行。知识蒸馏的训练时间很长，因为教师模型需要对全部预训练数据进行前向预测，以生成激活值或中间表示。

Rogers 等人 [176]列举了大量针对 BERT 的参数规模缩减研究，并报告了参数规模和计算时间的缩减情况以及由此产生的性能。对于若干种方法，内存和计算开销显著降低，而性能几乎不变。

3.5.6 总结

模型参数的数量、训练数据的规模以及训练所需的计算开销，是决定模型

性能的因素。Kaplan 等人 [102] 通过实验表明，增加参数数量和训练集规模能够可靠地提升性能，并给出了这种依赖关系的详细公式。如果计算预算固定，应使用非常大的模型和大量数据。

专家混合通过增加参数数量而无需增加计算开销来遵循这种方法。它们将输入路由到特定子网络，从而能够比单体网络取得更好的性能。例如，GLaM、悟道2.0 和 Switch 都采用了这种方法。然而，这些网络拥有数千亿个参数，并且需要专门的并行计算基础设施。

通常，训练好的网络过于庞大，需要缩减后才能适配较小的计算设备。一种可行的方法是低精度计算，它可以减少存储参数所需的内存。矩阵的低秩分解也能降低内存占用。最后，可以采用知识蒸馏来创建一个学生模型，使其模仿大型训练好的教师网络的内部工作方式。例如，DistilBERT 将内存占用减少了 40%，保留了原模型 99% 的性能，速度则提升了 60%。还有许多其他规模缩减方法也能取得类似结果。

3.6 特定应用的微调

在大型文本集合上对语言模型进行自监督预训练，随后对其进行微调以解决特定任务，已成为自然语言处理与理解领域的标准范式。研究表明，BERT 等预训练语言模型具有出色的泛化能力，并且可以轻松微调以完成多项任务。然而，有时仅针对特定领域任务进行简单微调并不足够，还需要采用其他迁移学习方法，以使模型更好地适应数据中的领域偏移 [166]。已有多篇综述深入探讨了迁移学习 [230, 252, 260]

微调会更新模型的所有层，包括嵌入层，但较高层的变化更大 [133]。首先，我们讨论微调是否会破坏预训练期间获得的知识。标准微调 以较低的计算开销，使用规模相对较小的微调训练数据集，对参数众多的大型预训练 PLM进行适配。我们将探讨这一阶段是否会出现过拟合。后续章节将介绍不同的微调方法：

- 中间微调 会先使用规模更大的训练集进行中间微调，然后再进行最终的目标微调。
- 多任务微调 通过同时针对多个任务进行微调，增强模型的能力。
- 冻结模型微调会针对微调任务调整一个额外的小型层，而不是改变大型预训练模型的所有权重。

- 创建少样本指令提示旨在为大型自回归预训练语言模型（如 GPT-3）生成输入，以通过零样本或少样本方法解决任务。

3.6.1 微调的性质

预训练语言模型通常通过监督训练进行微调，以适配特定任务。这种将模型从源任务迁移到相关目标任务的做法也称为迁移学习。如果我们有充足的自监督学习训练数据（未标注文本通常就是如此），而目标任务的标注数据很少，那么迁移学习尤其有益。Zhuang 等人 [260] 对迁移学习进行了综述。微调有许多优点：

- 模型通过利用预训练数据所提供的内容，获得关于语言及其句法和语义的详细知识。
- 预训练模型可以轻松适配新任务，例如添加一层简单的分类器。预训练模型的语言表示有助于微调，并且在此过程中只会发生轻微变化。
- 即使数据集很小，微调的性能也远胜于直接在有限数据上训练分类器。

像 BERT 这样的自编码器模型通常会针对分类任务进行微调，此时需要移除用于掩码语言建模和下一句预测的逻辑分类器。使用 [*CLS*] 词元或其他词元作为输入，新逻辑分类器以及所有模型参数都会与新任务一起进行端到端训练，持续若干轮（第 2.1.3 节）。与预训练相比，微调的开销相对较低。通常，只需投入预训练所需训练开销的一小部分，就能取得良好结果。

Tripuraneni 等人 [210] 从理论上证明，迁移学习所需的数据远少于单独学习各项任务所需的数据。他们证明，任务多样性提高时，迁移学习的效果也会提升。Bansal 等人 [7] 研究了使用预训练嵌入微调分类器的理论性质。作者证明，与标准分类器相比，这些分类器在训练集和测试集准确率之间的泛化差距更小。

灾难性遗忘

问题在于，微调是否会破坏模型原有的能力。这意味着，预训练模型经过几轮微调后，可能会丧失预训练后所具备的预测性能。一种可能的原因是灾难性遗忘，即所有参数都针对新的学习任务进行了调整，同时却遗忘了已学习的内容。

Merchant 等人 [133] 对 BERT 进行微调$_{\text{BASE}}$，采用了三种不同的任务：1）MNLI 句对分类任务 [229]，用于测量第一个句子是否蕴含第二个句子；2）SQuAD 问答 [173]，，需要在文本中标出问题的答案；3）依存句法分析 [50]，用于捕捉句子的句法结构。随后，他们考察了若干探测分类器在微调前后的性能。结果表明，微调模型检测语言学概念的准确率仅略有下降。MNLI 任务造成的降幅在大多数情况下不到 1%，而 SQuAD 和依存句法分析的降幅较大（但不到 3%）。因此，没有观察到灾难性遗忘。作者指出，微调主要改变了 BERT 的顶层，而依存句法分析也会影响更深层。 [216]提供了更详细的结果。

只有当两个任务之间存在相似性时，微调才能从预训练中获益。因此，预训练应采用一种损失函数，以促使模型学习单词、短语和文档层面的语义。此外，训练文档应来自与微调任务相近的领域，否则词汇表中可能不包含许多领域特定词。结果，领域特定词会被拆分成多个词元，从而妨碍模型学习并降低其在下游任务中的性能。接下来的章节将讨论能够提升 BERT 能力的替代训练方案。

微调与过拟合

在预训练期间，BERT 的参数会根据预训练数据进行调整，从而获得通用的语言表示。由于预训练提供了良好的初始化，因此只要不过度降低微调误差，就能避免在小型微调数据集上过拟合。

由于预训练语言模型的参数数量非常庞大，因此存在在微调数据上过拟合的风险。结果是，模型对未见数据的泛化能力可能较差，因而可能需要采取应对策略。D'Amour [42] 对这一欠规范性现象进行了全面讨论。Jiang 等人 [95]提出了一种正则化方法，使模型对输入的微小扰动保持不变，从而令局部邻域内的变化更加平滑。他们开发了一类 Bregman 近端点优化方法，通过惩罚模型在每次迭代中的大幅更新来实现正则化。Aghajanyan 等人 [2] 提出了表示坍塌的概念，指出微调模型会失去泛化能力。他们基于信赖域理论提出了微调优化方法，以较低成本缓解表示坍塌；与其他近期提出的微调方法相比，该方法所需成本仅为其一小部分，并且例如改进了在 GLUE 上微调 RoBERTa 的已知最佳结果。

使用多个随机种子对同一模型进行微调，可能会导致任务性能出现较大方差。大多数论文认为，这种现象是由灾难性遗忘以及微调数据集规模较小造成的。然而，Mosbach等人 [140]指出，微调常常会因梯度消失而面临优化问题。

此外，模型即使具有与成功模型相同的微调损失，也常常可能无法很好地泛化。这表明了上文提到的欠指定问题。作者建议使用较小的学习率并进行偏差校正，以避免训练早期出现梯度消失。此外，他们还建议增加微调的迭代次数。Rogers 等人 [176]提供了更多改进微调的方法。

3.6.2 微调变体

两阶段微调

中间训练集应与最终任务更为接近。尽管这种方法在某些情况下可以提升性能，但实验评估表明，在44%的情况下，性能反而下降 [163]。一项大型实验表明，采用要求进行高层推理并具备推理能力的任务进行中间训练，往往效果最佳 [165]。然而，作者也观察到预训练能力出现了灾难性遗忘。Gururangan 等人 [71] 表明，使用领域特定数据进行第二阶段预训练，在高资源和低资源场景下都能显著提升性能。此外，使用任务特定的未标注数据进行预训练，可以提升模型在各种任务和领域上的性能。

多任务微调

针对每项任务，都会在底层预训练模型上添加一个任务专用层。随后，使用所有任务同时训练模型。然而，有时性能并不会比标准微调有所提升 [141],，这或许是因为各项任务的要求相互矛盾。作为替代方案，可以根据相似度度量，从可用数据集中选择一部分微调任务 [131]。

HyperGrid [208] 是一种在 T5模型上进行评估的多任务学习方法。它学习网格级投影，以帮助针对不同任务对权重矩阵中的区域进行专门化。例如，单个模型可以同时适应 GLUE 和 SuperGLUE 的所有任务。尽管任务数量众多，该模型在 SuperGLUE 上的性能仍略优于单独模型。

用于加速微调的元学习

在微调过程中，需要将预训练的PLM调整为适用于新的自然语言处理任务。通常情况下，模型会在带标注的微调数据集上训练两到三个训练轮次。尽管这比在大型训练语料库上对模型进行预训练快得多，但仍需要付出

大量努力。为了减少这项工作，研究人员尝试通过元学习来准备预训练模型以进行微调。Yin [242]提供了元学习综述。

通常，有一组相关的微调任务 T_i。在元训练期间，从分布 $p(\mathrm{T})$中采样一个任务T_i 。随后，使用来自T_i^{train}的 K 训练样本训练模型，再在 T_i^{val}的验证集上进行测试。 T_i 的验证误差用作当前迭代中元学习框架的训练误差。

MAML 算法 [58] 遵循以下模式：

- 复制初始模型参数 $\boldsymbol{w}$的 $\boldsymbol{w}^{[i]}$ 。
- 使用 K 次梯度更新在训练集 $T_{\text{train}i}$ 上训练模型：$\hat{\boldsymbol{w}}^{[i]} \leftarrow \boldsymbol{w}^{[i]} - \gamma \partial L_i(\boldsymbol{w}^{[i]}, T_{\text{train}i})/\partial \boldsymbol{w}$
- 使用更新后的参数$\hat{\boldsymbol{w}}^{[i]}$ 在验证集 $T\text{val}_i$ 上应用模型。
- 使用验证集$\boldsymbol{w} \leftarrow \boldsymbol{w} - \beta \partial L_i$上的损失更新初始模型参数 $\boldsymbol{w}$：$(\hat{\boldsymbol{w}}^{[i]}, T_{\text{val}i})/\partial \boldsymbol{w}$

该方案应用于 BERT [6]。作者从未标注文本中收集待掩码的词元，并从少量词汇项中生成大量且丰富的元学习任务分布。在 17 项自然语言处理任务上，他们表明，与先进行语言模型预训练、再进行微调相比，这种元训练能带来更好的少样本泛化。Chen 等人 [28]为这些方法给出了依赖于数据的泛化界。

通过适配器微调冻结模型

用于任务适配的微调有一个缺点：每项任务都需要新的模型参数。任务适配器 [84] 旨在缓解这一问题。作者引入了适配器层，将其插入编码器块中，位置在多头注意力和前馈层之后（2.7）。现在，要将 Transformer 模型微调以适应新任务，无需重新学习所有参数，只需冻结网络的所有权重，保留适配器层和归一化层的权重即可。在 GLUE 等任务上，这种做法在保持模型质量的同时，显著减少了需要训练的参数。

Stickland 等人 [197]提出训练一个多任务学习版本的 BERT，而不是为不同任务使用多个适配器；该模型可同时用于多个任务。他们在 BERT 编码器块中添加低维投影注意力层作为旁路，将输入连接到层归一化层和后续的层归一化层。训练期间，他们根据各训练集的大小按比例从不同任务中抽取数据，并使用退火机制，使训练结束时各任务的训练样本分布趋于均衡。他们的结果优于 $\text{BERT}_{\text{BASE}}$ 模型的结果。

MAD-X [160] 是一个将多语言模型适配到任意语言和任务的框架。作者引入了特定于语言和任务的适配器，它们由一个将输入线性降维到小型向量的层、一个ReLU激活和一个线性升维层组成。特定于语言的适配器使用掩码语言模型目标进行训练，而模型的其余部分保持Frozen状态。特定于任务的适配器使用任务-

特定数据进行训练，同时固定其余参数。最后，在输入嵌入层之后和输出嵌入层之前添加可逆适配器，以减轻多语言词汇表与目标语言词汇表之间的差异。MAD-X在多种不同语言的命名实体识别和常识推理任务上达到了最先进水平。

LoRA [85] 冻结预训练模型的权重，并为模型添加可训练的旁路结构。这些旁路结构由可训练的矩阵变换组成，可将向量转换为低秩表示并还原为全秩表示。这样可以大幅减少可训练参数的数量（GPT-3 为原来的 1/30，GPT-2 为原来的 1/100），同时在许多自然语言处理任务中取得比传统微调更好的结果。*AdapterHub* [161] 是一个适配器仓库，截至撰写本文时，其中包含约 380 个适配器。AdapterHub 构建于 Hugging Face Transformer 库之上，可与现有的 Transformer 模型兼容。

微调 GPT-3

GPT-3 是一个极其强大的基础模型，但并未公开提供（第 3.1.2 节）。通过使用 API，利用用户特定的数据对 GPT-3 进行微调， [123],可以使模型适应特定领域的语言和特定任务。与下文所述的少样本示例和提示设计相比，这通常能够获得更高的质量。对于一个包含 1M 个词元、训练 4 个轮次的 175B 参数模型，OpenAI 收费约 120 美元。微调可以通过多种方式使用 [123]:

- 文本补全：为提示生成文本补全。
- 搜索：给定一个搜索查询和一组文档或标签，模型会根据每个文档与查询之间的语义相似度为其评分并进行排序。
- 分类：输入是一个查询和一组带标签示例，例如 [“我感觉棒极了”，“正面”]。然后，GPT-3 会预测该查询最有可能对应的标签。这可以像 BERT 一样用于任何类型的分类任务。
- 回答：输入是一个问题、一组包含背景信息的文档以及一些示例。模型会根据文档和示例中的信息生成回答。这类似于问答中的阅读理解任务（第 6.2 节）。
- 微调：使 GPT-3 适应特定领域的文本。
- 嵌入：获取输入文本的上下文嵌入向量，以便进行进一步处理或探索。

可以推测，GPT-3 和 PaLM 等其他基础模型若以这种方式进行微调，凭借其全面的语言知识，将提升许多领域的最先进水平。

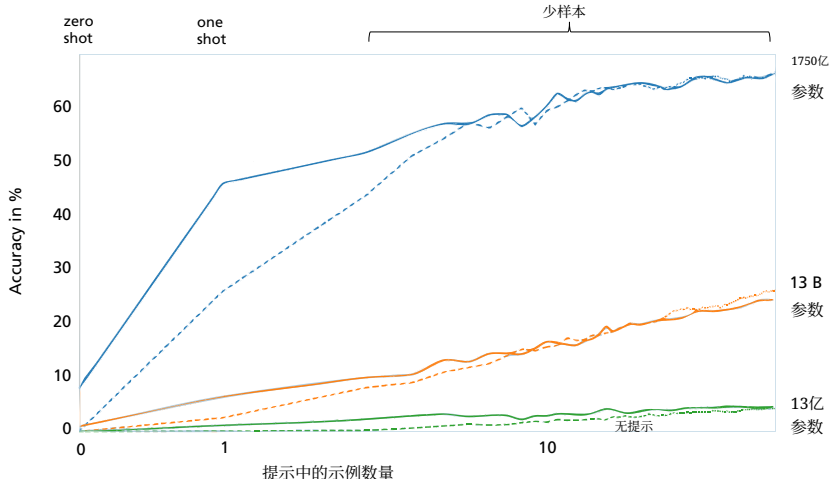


图 3.22 通过扩大模型规模以及增加提供的示例数量，GPT-3 的少样本学习准确率得以提升 [25]。任务是从一个单词中删除随机符号。任务的自然语言描述尤其能在单样本场景中为模型提供支撑。经作者慷慨许可转载的图像 [25, 第 4] 页。

3.6.3 创建少样本提示

对于 零样本学习，模型只会收到任务描述或提示，例如“将英语翻译成法语：奶酪 =>”，然后直接生成回答“*fromage*”。对于单样本 或少样本学习，模型会收到任务描述以及一个或多个示例，例如“将英语翻译成法语：海獭 => *loutre de mer*；奶酪 =>”，这有助于模型找到回答“*fromage*”。这一过程无需训练，不会改变模型的参数，模型会根据预训练期间获得的知识生成回答。

通过这种方式，可以用自然语言提示指示 GPT-3 生成短篇故事、歌曲、问题的答案、新闻稿、技术手册等 [181]。它还能使生成的文本适应特定的风格、个性或意识形态。以下是一些推荐用于少样本学习的提示 [150]:

- 摘要生成：模型接收一个长篇故事和提示“太长不看：”。
- 语法纠正 “原文：她没有去市场。标准美式英语：”

- 翻译： “英语：我不会说法语。法语：*Je ne parle pas français.* 英语：*Whereisthe restroom?*” 法语：
- 为论文生成大纲：“为一篇关于沃尔特·迪士尼及其对动画贡献的论文创建大纲：*I*：引言”

图 3.22 展示了不同 GPT-3 模型规模和不同给定示例数量下“少样本学习”的准确率。

在一篇全面的综述中，Liu 等人 [125]汇总了提示设计方法，用于为语言模型创建提示，使其能够可靠地生成所需的响应。例如，当我们想要识别文本“我今天没赶上公共汽车。”的情感时，可以插入提示“我感觉真是太 ”，并使用语言模型填入空白。提示分为两种类型：完形填空提示 [159], ，通过类似 BERT 的自编码器模型填充文本字符串中的空白；以及前缀提示 [117], ，通过自回归语言模型续写文本。

例如，在提示挖掘中 [96], ，会收集大量包含短语x 和 y 的句子。随后，利用 x 与 y之间的词语，或利用句法分析器生成的依存路径来生成提示。另一种方法是改述已有提示，例如将其翻译成另一种语言，再回译。通过基于梯度的搜索 [192]可以提高所需答案的概率，*AutoPrompt* 模型展示了这一做法。其他方法见 [62, 245]。不过需要注意的是，如果对手添加一些新提示 [79]，使用少样本提示指示的模型，其输出很容易被改变。

与其改进提示词元，让语言模型据此生成所需输出，不如优化一些“虚拟”词元的输入嵌入，从而生成所需答案。保持语言模型的参数不变时，可以通过梯度下降优化这种“连续”提示的嵌入 [121]。Lester 等人 [117] 将这一方法应用于 T5 Transformer，使用由 100 个词元组成的连续提示序列。在*SuperGLUE* 基准测试中，他们取得了 90.5% 的性能，与微调 T5 的效果相同。这表明，提示调优已具备与微调相竞争的实力，并且远胜于少样本指令。请注意，提示调优所需的工作量远低于微调，因为其参数数量少得多。若能将这一技术应用于 GPT-3 或 PaLM 等近期的自回归模型，将会很有意思。

3.6.4 用于推理少样本学习的思维链

为了提升语言模型的推理能力，提示中可以包含思维链，即一系列简短的句子，用来模仿人回答问题时可能采用的推理过程 [226]。图 2.21 展示了两个示例。其思路是，思维链让语言模型能够将多步问题拆分为若干中间步骤，并逐一求解，而不是一次性解决整个多步问题。

这种方法有许多优点。首先，思维链方法使模型能够将复杂推理任务分解为更简单的中间步骤，而模型可以逐步解决这些步骤。要解决一整类问题，只需提供少量思维链。其次，当模型执行中间步骤时，更容易检查模型在哪里出错。这

可能为改进思维链提供线索。思维链推理可应用于符号操作、常识推理和数学任务，并有望适用于人类能够通过语言解决的任何任务。

提示也不必局限于输入输出对或解释，还可以涵盖许多论据，包括应避免的事项、经验法则、推理链、正例或负例。Mishra 等人 [138] 研究了面向众包工作者的指令，其中包含如何完成任务的非常详细的说明。他们汇编了一个包含任务、指令和生成的输入输出对的数据集。随后，他们考察模型对类似任务的泛化能力。结果表明，若以对未见任务的泛化能力为评估标准，预训练语言模型会受益于指令（提升 19%）。不过，仍有很大的改进空间。

Du 等人 [52] 从理论上研究了少样本学习。他们考察了这样一种情况：模型先在若干任务的大规模训练集上进行预训练，随后再针对相关任务进行微调。他们从理论上推导出微调任务所需样本量的界限；若存在良好的共同表示，该样本量便可以减少。

3.6.5 微调模型以执行指令

与其通过少样本指令查询自回归预训练语言模型，不如对这些模型进行微调，使其无需额外示例即可执行指令。

InstructGPT [151] 是 GPT-3 的新版本。它经过优化，可以跟随指令，而不是预测最可能出现的下一个词语。GPT-3 现在无需一系列示例即可直接执行指令，例如：“写一个关于月亮和星星的短篇故事：”，模型会生成一个合理的故事。首次试验中，研究人员收集了一个包含 13,000 对指令和文本补全的数据集，用于调整 GPT-3。GPT-3 使用这些数据进行了微调。然而，该模型未能充分符合预期的人类偏好。因此，研究人员采用了不同的训练方法对模型进行了修改。

为了调整 GPT-3，研究人员采用了带有人类反馈的强化学习 方法。近端策略优化 (PPO) [186]遵循策略梯度模式。它近似表示在当前关于环境状态的观察 $s_t \in \mathrm{S}$ 和参数向量 $\boldsymbol{w}$ 条件下，在步骤 t 执行动作 $a_t \in \mathrm{A}$ 的条件分布 $\pi(a_t|s_t; \boldsymbol{w})$。在通常的强化学习中，环境会生成奖励，而算法则尝试最大化奖励的加权总和。该优化的梯度（策略梯度）可以从模型中轻松计算得到。PPO 在每个步骤计算一次更新，使代价函数最小化，同时确保与之前策略的偏差相对较小 [186]。

算法需要一个数值得分来衡量每个生成序列的质量。为了减少优化所需的数据，人工可以表达对轨迹 [198] $(\tau = , \boldsymbol{x})$ 的偏好，其中轨迹对应于指令 $\boldsymbol{x}$ 和生成文本 $\boldsymbol{y}$ 的成对组合。简单来说，目标是生成得到人工偏好的轨迹，

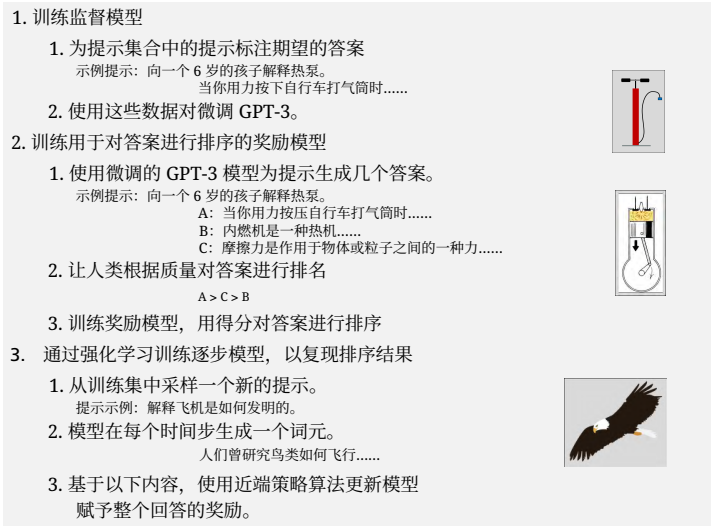


图 3.23 InstructGPT 的训练分为三个步骤 [151, 第 3] 页。首先，在指令及相应的文本补全上对 GPT-3 进行微调。随后，通过优化针对某项指令选择文本补全的过程，生成奖励模型。最后，训练一个策略，以逐词元生成具有最大奖励的回答。图像各部分的来源见表 A.1。

同时尽可能少地查询人工。为实现这一目标，假定存在一个奖励函数 $r(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{R}$，其具有以下性质：如果 $r(\boldsymbol{y}^{[1]}, \boldsymbol{x}^{[1]}) > r(\boldsymbol{y}^{[2]}, \boldsymbol{x}^{[2]})$，则 $(\boldsymbol{y}^{[1]}, \boldsymbol{x}^{[1]})$ 优于 $(\boldsymbol{y}^{[2]}, \boldsymbol{x}^{[2]})$。原始策略 $\pi(a_t|s_t; \boldsymbol{w})$ 诱导出条件分布 $\pi(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w})$。为构造该分布，奖励函数$r(\boldsymbol{y}, \boldsymbol{x})$ 由带有参数 $\boldsymbol{u}$ 的深度神经网络 $\hat{r}(\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{u})$ 进行近似。该网络通过三个交替步骤进行训练（图 3.23）：

1. 使用策略 $\pi(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w})$ 生成一组轨迹 $\{\tau^1, \ldots, \tau^i\}$。为最大化奖励 $\hat{r}(\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{u})$，通过强化学习更新参数 $\boldsymbol{w}$ 。2. 从 $\{\tau^1, \ldots, \tau^i\}$ 中选择轨迹对 $(\sigma^{[1]}, \sigma^{[2]})$，并提交给人工进行比较。3. 对奖励函数 $\hat{r}(\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{u})$ 的参数 $\boldsymbol{u}$ 进行优化，使其与截至目前从人工收集到的比较结果相对应。

对于一组包含 33k 条指令的数据，构建了一个具有 6B 个参数的奖励模型 $\hat{r}(\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{u})$，其中 $\boldsymbol{x}$ 是指令， $\boldsymbol{y}$ 是文本补全 [198]。它会从一小组候选文本补全中选出最佳文本补全。近端策略优化（PPO）被用作强化模型 [151, p。 41]。为避免灾难性遗忘（第 3.6.1 节），预训练样本被混入微调过程。

随后，通过又一次强化学习步骤应用奖励模型，生成最终模型。在此过程中，InstructGPT 会针对一条说明生成文本补全。奖励模型会计算奖励，并更新策略，以

近似奖励模型中编码的偏好。通过模仿人类话语，模型会隐式地学习人类意图和偏好。这一过程称为 对齐人类偏好 ，Askell 等人对此进行了详尽讨论 [5]。

InstructGPT 结果

将具有 1750亿参数的 GPT-3 模型通过监督方式在 1.3 万个指令完成示例上进行微调，并将其作为名为 SFT 的基础模型。随后，人工评估者再次对最终的文本补全进行评分 [151]。在 85% 的情况下，InstructGPT 的文本补全优于标准 GPT-3 的输出；在 71% 的情况下，优于少样本 GPT-3 的输出。

具体而言，评估者发现，InstructGPT 在 92% 的情况下会尝试遵循正确的说明，而 SFT 为 85%，少样本 GPT-3 [151,p 为 75%。 53]。此外，InstructGPT 在 50% 的情况下遵循显式约束，而 SFT 为 43%，SFT 为 34%，少样本 GPT-3 为 28%。在 InstructGPT 的 20% 的案例中观察到了幻觉，而 SFT 为 16%，少样本 GPT-3 为 50%。最后，评估者发现，InstructGPT 在 92% 的案例中使用适合客户助理的语言；SFT 约为 90%，GPT-3 少样本约为 85%。InstructGPT 还在若干自然语言基准测试上接受评估，其结果与 GPT-3[151, p 非常相似。 56]。

事实证明，InstructGPT 能够泛化到未见过的标注者偏好。因此，InstructGPT 并非只是适应少数训练标注者的偏好。此外，InstructGPT 生成的有害语言略少于标准 GPT-3。然而，InstructGPT 仍会犯一些简单的错误，例如，在面对带有错误前提的说明时，模型有时会错误地假设该前提为真。请注意，结果取决于标注者的主观偏好。

对不同选项进行比较并不一定是产生改进信号的最有效方法。例如，可以要求标注者编辑模型响应，使其变得更好，或者用自然语言生成对模型响应的批评。此外，为标注者设计向语言模型提供反馈的界面存在极其广阔的选择空间；这是一个有趣的人机交互问题。作者指出，上述将 GPT-3 与人类偏好对齐的成本，仅占训练 GPT-3 所花费成本的 1.6%。因此，与仅仅扩大模型规模相比，投入更多精力进行对齐似乎更为合理。

结果表明，InstructGPT 技术可能会让语言模型更有帮助、更真实且更无害。从某种意义上说，InstructGPT 就像一个智能助手，能够生成语音并提供信息。然而，目前该模型并不适用于安全关键应用，因为无法排除它出现故障的可能性。目前仍缺乏类似 Gopher 或 PaLM（第 3.1.2 节）的全面评估，以展示这种方法的实际效用。可以预期，将这种方法与

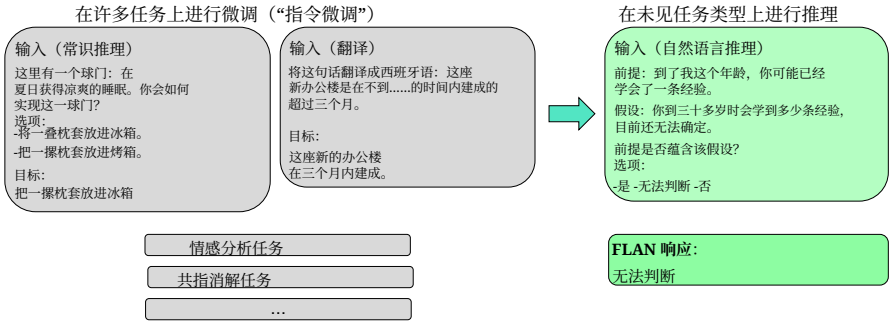


图 3.24 FLAN 指令微调在一组任务上对预训练语言模型进行微调，每项任务都配有十种不同模板的指令（左）。根据这些模板构造提示后，训练好的模型便可应用于未见过的任务（右）。图片经作者许可改编自 [227, 第 1]页。

WebGPT（第 6.2.3 节）和 Retro（第 6.2.3 节）所采用的检索技术相结合，将提升 InstructGPT 的性能、可靠性和正确性。

使用 FLAN 进行指令微调

FLAN [227] 使用指令微调来提升语言模型响应自然语言提示的能力。语言模型必须通过监督学习，学会执行提示所描述的任务并遵循指令，即使面对不熟悉的任务也不例外（图 3.24）。作者将 62 个公开可用的自然语言处理数据集归为十二个任务簇，例如“情感”“自然语言推理”“摘要”等。对于每个数据集，他们都编写了十种以自然语言描述任务的模板。随后，对现有语言模型进行微调，使其能够更好地回答提示。

该方法应用于一个拥有 137B 个参数的 LaMDA-PT 语言模型，并结合检索和过滤器（第 6.6.3 节）。在 18 项自然语言推理任务中，研究人员将 FLAN 模型与 LaMDA-PT 137B、GPT-3 175B 和 GLaM 64B 进行了比较。在 18 项任务中的 14 项里，FLAN 显著提升了其未经修改的对应模型的性能，结果也优于竞争模型；但在 4 项任务中，表现不及 GLaM [227]。在若干任务中，FLAN 的表现甚至超过了少样本 GPT-3，优势很大。

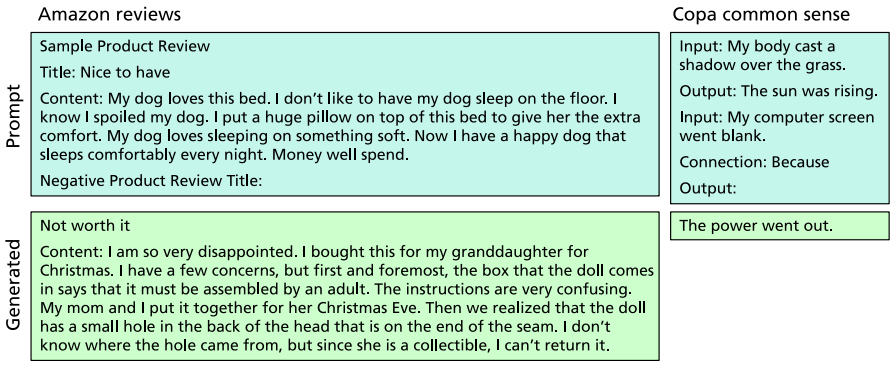


图 3.25 利用少样本 UDG 策略，GPT-3 和其他基础模型可以生成新数据。这里展示了两个示例的提示及生成的答案：亚马逊评论和 COPA 常识推理 [225]。

3.6.6 基础模型生成带标签的数据

GPT-3 和其他基础模型的少样本学习性能，使其能够为其他模型生成新的高质量训练数据。通过无监督数据生成 （*UDG*），可以为下游任务的模型创建微调数据；否则，这些数据需要通过人工标注来生成。该方法与第 4.2.3 节所述方法类似。

数据生成的思路是利用语言模型，根据任务描述和少量输入-标签样例对来学习输入-标签关系 [225]。对于分类任务，语言模型不再生成并预测标签，而是以输出类别和任务描述作为输入，创建输入文本。对于亚马逊产品评论这类分类任务，该方法能够为每个类别生成 1 万个新示例，覆盖范围远大于当前可用的标注数据。事实证明，最多 32 个少样本示例仍能提高所生成训练数据的质量。示例见图 3.25。作者使用了一个额外模块来过滤噪声样例。在该方法中，如果训练好的分类器无法以高概率将给定训练样本识别为其标签，则会移除该样本。

在经UDG数据增强的SuperGLUE数据上进行微调的T5-XXL编码器-解码器模型，能够将自然语言理解SuperGLUE任务的整体准确率提升至90.4%，甚至能够以90.3%的成绩击败DeBERTa。此外，该方法在一系列文本分类和情感分析任务上也取得了非常高的性能得分 [225]。

3.6.7 总结

在大规模文本集合上对基础模型进行预训练，随后在小型标注数据集上进行监督式微调时，预训练语言模型在许多自然语言处理任务上取得了前所未有的性能。研究表明，微调只会略微改变模型参数，而且通常不会发生灾难性遗忘。通常，如果在几个训练轮次后停止微调，就不会观察到过拟合。如有必要，也可以采用一些方法来避免过拟合。

微调可以通过不同方式执行。有人建议，在对小型数据集进行最终微调之前，先使用相关性更高的数据集进行中间微调。此类方法的结果褒贬不一。此外，还可以同时针对多个任务进行微调。在某些情况下，这可能会提升性能。另一种方法是通过元学习来加速微调。为了避免更改完整模型，可以定义适配器层，仅调整其参数。这样可以大幅减少可训练参数的数量，同时仍能在微调任务上取得良好性能。最后，近期已经为 GPT-3 等专有模型提供了微调 API。

GPT-3 和 PaLM 等基础模型可以通过提示来执行特定任务，无需训练。人们收集了大量不同的提示，用于指示模型完成任务。InstructGPT 是 GPT-3 的新版本，能够直接接受指令，并为广泛的任务提供答案。该模型通过强化学习适应用户评价，从而经过定制以执行这些指令。指令微调是一种变体，在这种方法中，基础模型经过微调，为各种任务的指令提供更优答案。事实证明，此后模型即使面对未见过的任务，也能生成更好的答案。

最后，可以使用大型语言模型生成高质量的训练数据，用于微调。同样，少样本学习技术可用于为特定学习任务生成输入文本。这样一来，便能扩充稀缺的训练数据，并取得更好的微调结果。

参考文献

[1] O. Agarwal、H. Ge、S. Shakeri 和 R. Al-Rfou。《基于知识图谱的知识增强语言模型预训练合成语料生成》。2021年3月13日2010 12688。arXiv：。。[2] A. Aghajanyan、A. Shrivastava、A. Gupta、N. Goyal、L. Zettlemoyer 和 S. Gupta。《通过减少表征坍塌改进微调》。2020年8月6日。arXiv：2008.03156。[3] J. Ainslie、S. Ontanon、C. Alberti、P. Pham、A. Ravula 和 S. Sanghai。《ETC：在 Transformer 中编码长篇结构化数据》。2020年。arXiv：2004.08483。[4] A. Alvi。使用 *DeepSpeed* 和 *Megatron* 训练 *Megatron-Turing NLG 530B*：全球规模最大、功能最强大的生成式语言模型。微软研究院。2021年10月11日。网址：
https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-

megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/（访问于2021年11月12日）。[5] Askell 等人。《作为对齐研究实验室的通用语言助手》。12月 2021 2112 00861 [ ] 。arXiv：。cs。[6] T. Bansal、R. Jha、T. Munkhdalai 和 A. McCallum。《面向少样本自然语言分类任务的自监督元学习》。2020年。arXiv：2009.08445。[7] Y. Bansal、G. Kaplun 和 B. Barak。《自监督学习中，理性必然意味着泛化，且可证明》。2020年。arXiv：2010.08508。[8] H. Bao 等人。《UniLMv2：用于统一语言模型预训练的伪掩码语言模型》。载于：国际机器学习会议。PMLR，2020年，第642–652页。[9] A. Bapna 等人。为接下来的一千种语言构建机器翻译系统。2022年5月16日。arXiv：2205.03983 [cs]。[10] I. Beltagy、M. E. Peters 和 A. Cohan。《Longformer：长文档 Transformer》。2020年。arXiv：2004.05150。[11] 基准测试。GLUE 基准。2021年8月5日。网址：https://gluebenchmark.com/（访问于2021年8月5日）。[12] Y. Bengio、A. Courville 和 P. Vincent。《表示学习：综述与新视角》。载于：IEEE 模式分析与机器智能汇刊 35.8（2013年），第1798–1828页。[13] J. Berant、A. Chou、R. Frostig 和 P. Liang。《基于问答对的 Freebase 语义解析》。载于：2013年自然语言处理经验方法会议论文集。EMNLP 2013。美国华盛顿州西雅图：计算语言学协会，2013年10月，第1533–1544页。网址：https://aclanthology.org/D13-1160（访问于2021年12月14日）。[14] M. Bevilacqua 和 R. Navigli。《突破80%的玻璃天花板：通过整合知识图谱信息提升词义消歧的最先进水平》。载于：计算语言学协会论文集，2020年，第2854–2864页。[15] C. Bizer、J. Lehmann、G. Kobilarov、S. Auer、C. Becker、R. Cyganiak 和 S. Hellmann。《DBpedia——数据网络的结晶点》。载于：Web 语义学杂志 7.3（2009年），第154–165页。[16] S. Black、G. Leo、P. Wang、C. Leahy 和 S. Biderman。GPT-Neo：使用 Mesh-Tensorflow 进行大规模自回归语言建模。Zenodo，2021年3月21日。doi：10.5281/zenodo.5297715。[17] O. Bojar 等人。《2014年统计机器翻译研讨会的研究结果》。载于：第九届统计机器翻译研讨会论文集，2014年，第12–58页。[18] K. Bollacker、C. Evans、P. Paritosh、T. Sturge 和 J. Taylor。《Freebase：一个协作创建、用于组织人类知识的图数据库》。载于：2008年 ACM SIGMOD 数据管理国际会议论文集，2008年，第1247–1250页。[19] R. Bommasani 等人。《基础模型的机遇与风险》。2021年。arXiv：2108.07258。[20] A. Bordes、N. Usunier、A. Garcia-Duran、J. Weston 和 O. Yakhnenko。《用于多关系数据建模的嵌入翻译》。载于：神经信息处理系统进展 26 （ 2013 年），第 2787–2795页。[21] S. Borgeaud 等人。《通过从万亿级词元中检索来改进语言模型》。2021年12月8日。arXiv：2112.04426 [cs]。[22] A. Borzunov 等人。Petals：大型模型的协作推理与微调。2022年9月2日。doi：10.48550/arXiv.2209.01188。arXiv：2209.01188 [cs]。[23] G. Branwen。《GPT-3 创意小说》。载于：（2020年6月19日）。网址：https://www.gwern.net/GPT-3（访问于2021年11月14日）。[24] S. Brin 和 L. Page。《大型超文本网络搜索引擎的剖析》。载于：计算机网络与 ISDN 系统 30.1-7（1998年），第107–117页。[25] T. B. Brown 等人。《语言模型是少样本学习器》。2020年。arXiv：2005.14165。[26] J. Casper。Megatron-LM 和 Megatron-DeepSpeed 的这个分支是什么？BigScience 研讨会，2022年10月25日。网址：https:// github.com/ bigscience- workshop/ Megatron- DeepSpeed（访问于2022年10月25日）。

[27] D. Chen。《Openqa-Tutorial Danqi/Acl2020》。2020年7月5日。网址：https://github.com/danqi/acl2020- openqa-tutorial（访问于2021年2月24日）。[28] Q. Chen、C. Shui 和 M. Marchand。《元学习的泛化界：一种信息论分析》。载于：《神经信息处理系统进展》34（2021年）。[29] T. Chen、J. Frankle、S. Chang、S. Liu、Y. Zhang、Z. Wang 和 M. Carbin。《预训练BERT网络的彩票假设》。2020年。arXiv：2007.12223。[30] W. Chen、Y. Su、X. Yan 和 W. Y. Wang。《KGPT：用于数据到文本生成的知识支撑的预训练》。2020年。arXiv：2010.02307。[31] Z. Chi、L. Dong、S. Ma、S. H. X.-L. Mao、H. Huang 和 F. Wei。《mT6：结合翻译对的多语言预训练文本到文本转换器》。2021年。arXiv：2104.08692。[32] Z. Chi、L. Dong、F. Wei、W. Wang、X.-L. Mao 和 H. Huang。《通过预训练实现跨语言自然语言生成》。载于：AAAI。2020年，第7570–7577页。[33] R. Child、S. Gray、A. Radford 和 I. Sutskever。《使用稀疏Transformer生成长序列》。2019年。arXiv：1904.10509。[34] K. Choromanski 等人。《重新思考使用Performers的注意力机制》。2020年。arXiv：2009.14794。[35] A. Chowdhery 等人。《PaLM：使用Pathways扩展语言建模》。2022年4月5日。arXiv：2204。02311 [cs]。[36] P. F. Christiano、J. Leike、T. Brown、M. Martic、S. Legg 和 D. Amodei。《从人类偏好中进行深度强化学习》。载于：《神经信息处理系统进展》30（2017年）。[37] H. W. Chung、T. Févry、H. Tsai、M. Johnson 和 S. Ruder。《重新思考预训练语言模型中的嵌入耦合》。2020年。arXiv：2010.12821。[38] A. Clark 等人。《路由语言模型的统一缩放定律》。2022年2月9日。arXiv：2202。01169 [cs]。[39] K. Clark、M.-T. Luong、Q. V. Le 和 C. D. Manning。《ELECTRA：将文本编码器作为判别器而非生成器进行预训练》。2020年。arXiv：2003.10555。[40] A. Conneau、G. Lample、R. Rinott、A. Williams、S. R. Bowman、H. Schwenk 和 V. Stoyanov。《XNLI：评估跨语言句子表示》。2018年9月13日。arXiv：1809.05053。[41] A. Conneau 等人。《大规模无监督跨语言表示学习》。2020年4月8日 1911 02116。arXiv：。。[42] A. D'Amour。《欠规范性如何给机器学习带来挑战》。Google AI 博客。2021年10月18日。网址：http://ai.googleblog.com/2021/10/how-underspecification-presents.html（访问于2021年10月25日）。[43] Y. Dai、S. Wang、N. N. Xiong 和 W. Guo。《知识图谱嵌入综述：方法、应用与基准测试》。载于：Electronics 9.5（2020年），第750页。[44] Z. Dai、Z. Yang、Y. Yang、W. W. Cohen、J. Carbonell、Q. V. Le 和 R. Salakhutdinov。《Transformer-XL：具有长期依赖的语言建模，2019年》。载于：URL Httpsopenreview Netforum。2019年。[45] T. Dash、S. Chitlangia、A. Ahuja 和 A. Srinivasan。《将领域知识融入深度神经网络》。2021年。arXiv：2103.00180。[46] L. de Alwis、A. Dissanayake、M. Pallewatte、K. Silva 和 U. Thayasivam。《语义表格解读综述》。载于：（2018年7月13日）。网址：http://semantic-web-journal.org/system/ files/swj1946.pdf。[47] X. Deng、H. Sun、A. Lees、Y. Wu 和 C. Yu。《TURL：通过表示学习实现表格理解》。2020年12月3日。arXiv：2006.14806。[48] J. Devlin。mBERT——多语言BERT。GitHub。2019年。网址：https : / / github. com / google - research/bert/blob/master/multilingual.md（访问于2021年2月21日）。[49] J. Devlin、M.-W. Chang、K. Lee 和 K. Toutanova。《BERT：用于语言理解的深度双向Transformer预训练》。2018年。arXiv：1810.04805。[50] T. Dozat 和 C. D. Manning。《用于神经依存句法分析的深度双仿射注意力》。2016年。arXiv：1611.01734。[51] N. Du 等人。《GLaM：使用混合专家高效扩展语言模型》。2021年12月13日。arXiv：2112。06905 [cs]。

[52] S. S. Du、W. Hu、S. M. Kakade、J. D. Lee 和 Q. Lei。《通过学习表示进行少样本学习：可证明地》。2020年。arXiv：2002.09434。[53] Z. Du。GLM。THUDM，2021年12月14日。网址：https:// github.com/ THUDM/ GLM（访问于2021年12月17日）。[54] Z. Du、Y. Qian、X. Liu、M. Ding、J. Qiu、Z. Yang 和 J. Tang。《所有自然语言处理任务都是生成任务：一种通用预训练框架》。2021年3月。

18, 2021 2103 10360 arXiv：。

[ ]

cs。[55] Z. Du、Y. Qian、X. Liu、M. Ding、J. Qiu、Z. Yang 和 J. Tang。《GLM：采用自回归空白填充的通用语言模型预训练》。2021年11月1日。网址：https://aclanthology.org/2022.acl-long.26/（访问于2021年12月17日）。[56] W. Fedus、B. Zoph 和 N. Shazeer。《Switch Transformer：使用简单高效的稀疏性扩展至万亿参数模型》。2021年。arXiv：2101.03961。[57] F. Feng、Y. Yang、D. Cer、N. Arivazhagan 和 W. Wang。《语言无关的BERT句子嵌入》。2020年7月3日。arXiv：2007。01852 [cs]。[58] C. Finn、P. Abbeel 和 S. Levine。《模型无关元学习：快速适应深度网络》。载于：国际机器学习会议。PMLR，2017年，第1126–1135页。[59] Q. Fournier、G. M. Caron 和 D. Aloise。《更快、更轻量Transformer的实用综述》。2021年3月26日。arXiv：2103。14636 [cs]。[60] P. Ganesh 等人。《压缩大规模基于Transformer的模型：以BERT为例》。2020年。arXiv：2002.11985。[61] L. Gao 等人。《The Pile：用于语言建模的800GB多样化文本数据集》。2020年。arXiv：2101.00027。[62] T. Gao、A. Fisch 和 D. Chen。《让预训练语言模型成为更好的少样本学习器》。2020年。arXiv：2012.15723。[63] H. Gong、Y. Sun、X. Feng、B. Qin、W. Bi、X. Liu 和 T. Liu。《TableGPT：利用表格结构重构和内容匹配进行少样本表格到文本生成》。载于：第28届国际计算语言学会议论文集。2020年，第1978–1988页。[64] M. A. Gordon、K. Duh 和 N. Andrews。《压缩BERT：研究权重剪枝对迁移学习的影响》。2020年。arXiv：2002.08307。[65] J. Gou、B. Yu、S. Maybank 和 D. Tao。《知识蒸馏综述》。2021年1月26日 2006 05525。arXiv：。。[66] N. Goyal、J. Du、M. Ott、G. Anantharaman 和 A. Conneau。《用于多语言掩码语言建模的更大规模Transformer》。2021年。arXiv：2105.00572。[67] A. Grover 和 J. Leskovec。《Node2vec：面向网络的可扩展特征学习》。载于：第22届ACM SIGKDD国际知识发现与数据挖掘会议论文集。2016年，第855–864页。[68] A. Gu、K. Goel 和 C. Ré。《使用结构化状态空间高效建模长序列》。2021年。arXiv：2111.00396。[69] A. Gu、K. Goel 和 C. Ré。《带注释的S4》。2021年。网址：https://srush.github.io/annotated-s4/（访问于2022年4月5日）。[70] A. Gupta。《对角状态空间与结构化状态空间同样有效》。2022年。arXiv：2203.14343。[71] S. Gururangan、A. Marasović、S. Swayamdipta、K. Lo、I. Beltagy、D. Downey 和 N. A. Smith。《不要停止预训练：使语言模型适应领域和任务》。2020年。arXiv：2004.10964。[72] K. Guu、K. Lee、Z. Tung、P. Pasupat 和 M.-W. Chang。《REALM：检索增强语言模型预训练》。2020年。arXiv：2002.08909。[73] C. Hawthorne 等人。《通用长上下文自回归建模与Perceiver AR》。2022年。arXiv：2202.07765。[74] J. He、J. Qiu、A. Zeng、Z. Yang、J. Zhai 和 J. Tang。《FastMoE：一种快速的混合专家训练系统》。2021年3月24日。arXiv：2103。13262 [cs]。[75] P. He、J. Gao 和 W. Chen。《DeBERTaV3：使用ELECTRA风格的预训练和梯度解耦嵌入共享改进DeBERTa》。2021年。arXiv：2111.09543。[76] P. He、X. Liu、J. Gao 和 W. Chen。《DeBERTa：具有解耦注意力的解码增强BERT》。2021年1月11日。arXiv：2006.03654。

[77] W. D. Heaven。《这个无所不知的人工智能通过不停阅读整个万维网来学习》。《麻省理工科技评论》。2020年9月4日。网址：https : / / www . technologyreview . com / 2020 / 09 / 04 / 1008156 / knowledge - graph - ai - reads - web - machine - learning - natural - language - processing/（访问日期：2021年12月1日）。[78] K. M. Hermann、T. Kocisky、E. Grefenstette、L. Espeholt、W. Kay、M. Suleyman 和 P. Blunsom。《教机器阅读与理解》。2015年。arXiv：1506.03340。[79] A. Hern。《TechScape：人工智能的黑暗技艺渐入佳境》。载于：《卫报》。科技（2022年9月21日）。ISSN：0261-3077。网址：https://www.theguardian.com/technology/2022/sep/21/ais-dark-arts-come-into-their-own（访问日期：2022年10月1日）。[80] D. Hernandez、J. Kaplan、T. Henighan 和 S. McCandlish。《迁移缩放定律》。2021年2月1日。arXiv：2102.01293 [cs]。[81] J. Herzig、P. K. Nowak、T. Müller、F. Piccinno 和 J. M. Eisenschlos。《Tapas：通过预训练进行弱监督表格解析》。2020年。arXiv：2004.02349。[82] G. Hinton、O. Vinyals 和 J. Dean。《蒸馏神经网络中的知识》。2015年。arXiv：1503.02531。[83] J. Hoffmann 等人。《训练计算最优的大语言模型》。2022年。arXiv：2203.15556。[84] N. Houlsby 等人。《面向自然语言处理的参数高效迁移学习》。载于：国际机器学习会议。PMLR，2019年，第2790–2799页。[85] E. J. Hu、Y. Shen、P. Wallis、Z. Allen-Zhu、Y. Li、S. Wang 和 W. Chen。《LoRA：大语言模型的低秩适配》。2021年。arXiv：2106.09685。[86] J. Hu、S. Ruder、A. Siddhant、G. Neubig、O. Firat 和 M. Johnson。《XTREME：用于评估跨语言泛化的大规模多语言多任务基准》。载于：国际机器学习会议。PMLR，2020年，第4411–4421页。[87] Z. Hu、Y. Dong、K. Wang、K.-W. Chang 和 Y. Sun。《GPT-GNN：图神经网络的生成式预训练》。载于：第26届ACM SIGKDD 国际知识发现与数据挖掘会议论文集。2020年，第1857–1867页。[88] H. Huang、Y. Liang、N. Duan、M. Gong、L. Shou、D. Jiang 和 M. Zhou。《Unicoder：通过多项跨语言任务预训练实现的通用语言编码器》。2019年。arXiv：1909.00964。[89] A. Iyer。《GPT-3 的免费替代品 GPT-Neo 值得期待》。VentureBeat。2021年5月15日。网址：https://venturebeat.com/2021/05/15/gpt-3s-free-alternative-gpt-neo-is-something-to-be-excited-about/（访问日期：2022年1月3日）。[90] M. Iyyer、W.-t. Yih 和 M.-W. Chang。《基于搜索的神经结构化学习用于序列问答》。载于：第55届计算语言学协会年会论文集。第1卷，长篇论文。2017年，第1821–1831页。[91] G. Izacard 和 E. Grave。《利用段落检索与生成模型进行开放域问答》。载于：第16届欧洲计算语言学协会分会会议论文集。主会场。EACL 2021。在线：计算语言学协会，2021年4月1日，第874–880页。网址：https://www.aclweb.org/anthology/2021.eacl-main.74（访问日期：2021年6月16日）。[92] A. Jaegle、F. Gimeno、A. Brock、A. Zisserman、O. Vinyals 和 J. Carreira。《Perceiver：通过迭代注意力实现的通用感知》。2021年6月22日。arXiv：2103 . 03206 [cs，eess]。[93] A. Jaegle 等人。《Perceiver IO：面向结构化输入与输出的通用架构》。2021年8月2日。arXiv：2107.14795。[94] S. Ji、S. Pan、E. Cambria、P. Marttinen 和 S. Y. Philip。《知识图谱综述：表示、获取与应用》。载于：IEEE 神经网络与学习系统汇刊。（2021年）。[95] H. Jiang、P. He、W. Chen、X. Liu、J. Gao 和 T. Zhao。《SMART：通过原则化正则化优化实现稳健高效的预训练自然语言模型微调》。载于：第58届计算语言学协会年会论文集。ACL 2020。在线：计算语言学协会，2020年7月，第2177–2190页。doi：10.18653/v1/ 2020.acl-main.197。

[96] Z. Jiang、F. F. Xu、J. Araki 和 G. Neubig。《我们如何知道语言模型知道什么？》。载于：计算语言学协会会刊。8（2020），第423–438页。[97] X. Jiao 等人。《TinyBERT：为自然语言理解蒸馏BERT》。2019年。arXiv：1909.10351。[98] M. Joshi、D. Chen、Y. Liu、D. S. Weld、L. Zettlemoyer 和 O. Levy。《SpanBERT：通过表示和预测跨度改进预训练》。载于：计算语言学协会会刊。8（2020），第64–77页。[99] M. Joshi、E. Choi、D. S. Weld 和 L. Zettlemoyer。《TriviaQA：用于阅读理解的大规模远程监督挑战数据集》。2017年。arXiv：1705.03551。[100] D. Jurafsky 和 J. H. Martin。《语音与语言处理：自然语言处理、计算语言学与语音识别导论》。第3版草稿。2022年1月12日。[101] R. E. Kalman。《一种新的线性滤波与预测问题方法》。载于：（1960年）。[102] J. Kaplan 等人。《神经语言模型的缩放定律》。2020年。arXiv：2001.08361。[103] V. Karpukhin、B. Oğuz、S. Min、L. Wu、S. Edunov、D. Chen 和 W.-t. Yih。《面向开放域问答的密集段落检索》。2020年。arXiv：2004.04906。[104] K. Karthikeyan、Z. Wang、S. Mayhew 和 D. Roth。《多语言BERT的跨语言能力：一项实证研究》。2020年2月15日。arXiv：1912.07840。[105] A. Katharopoulos、A. Vyas、N. Pappas 和 F. Fleuret。《Transformer 即 RNN：具有线性注意力的快速自回归 Transformer》。载于：国际机器学习会议。PMLR，2020年，第5156–5165页。[106] P. Kharya 和 A. Alvi。《使用 DeepSpeed 和 Megatron 训练 Megatron-Turing NLG 530B：全球最大、最强大的生成式语言模型》。NVIDIA 开发者博客。2021年10月11日。网址：
https://developer.nvidia.com/blog/using- deepspeed- and-
m
e
g
a
t
r
o
n-
to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/（访问日期：2022年1月8日）。[107] T. N. Kipf 和 M. Welling。《基于图卷积网络的半监督分类》。2016年。arXiv：1609.02907。[108] N. Kitaev、L. Kaiser 和 A. Levskaya。《Reformer：高效的 Transformer》。2020年。arXiv：2001.04451。[109] T. Kwiatkowski 等人。《Natural Questions：问答研究基准》。载于：计算语言学协会会刊。7（2019），第453–466页。[110] G. Lai、Q. Xie、H. Liu、Y. Yang 和 E. Hovy。《RACE：来自考试的大规模阅读理解数据集》。2017年。arXiv：1704.04683。[111] G. Lample 和 A. Conneau。《跨语言语言模型预训练》。2019年。arXiv：1901.07291。[112] G. Lample、A. Sablayrolles、M. Ranzato、L. Denoyer 和 H. Jégou。《使用乘积键的大容量记忆层》。2019年。arXiv：1907.05242。[113] Z. Lan、M. Chen、S. Goodman、K. Gimpel、P. Sharma 和 R. Soricut。《ALBERT：用于语言表示自监督学习的轻量级BERT》。2020年。arXiv：1909. 11942。[114] J. Lee、M. Sung、J. Kang 和 D. Chen。《大规模学习短语的稠密表示》。2021年1月2日。arXiv：2012.12624。[115] O. Lehmberg、D. Ritze、R. Meusel 和 C. Bizer。《包含时间与上下文元数据的大型公开网页表格语料库》。载于：第25届万维网国际会议伴随论文集。2016年，第75–76页。[116] D. Lepikhin 等人。《GShard：通过条件计算与自动分片扩展巨型模型》。2020年。arXiv：2006.16668。[117] B. Lester、R. Al-Rfou 和 N. Constant。《规模对参数高效提示调优的力量》。2021年。arXiv：2104.08691。[118] M. Lewis、M. Ghazvininejad、G. Ghosh、A. Aghajanyan、S. Wang 和 L. Zettlemoyer。《通过改述进行预训练》。2020年。arXiv：2006.15020。[119] M. Lewis 等人。《BART：面向自然语言生成、翻译与理解的去噪序列到序列预训练》。2020年。arXiv：1910.13461。

[120] P. Li 等人. “用于药物发现的表达性分子全局表示学习的有效自监督框架”。载于：《Brief Bioinform》22.6（2021年11月5日），bbab109。ISSN：1477-4054。DOI：10.1093/bib/bbab109。PMID：33940598。[121] X. L. Li 和 P. Liang。“前缀调优：优化用于生成的连续提示”。2021年。arXiv：2101.00190。[122] O. Lieber、O. Sharir、B. Lentz 和 Y. Shoham。“Jurassic-1：技术细节与评估”。载于：（2021年），第9页。URL：https : / / uploads - ssl . webflow. com / 60fd4503684b466578c0d307 / 61138924626a6981ee09caf6_jurassic_tech_paper.pdf。[123] R. Lim、M. Wu 和 L. Miller。为你的应用定制 GPT-3 14, 2021 3 02 16 2022。OpenAI。12月。。URL：https://openai.com/blog/customized-gpt- /（访问日期：/ /）。[124] X. V. Lin、R. Socher 和 C. Xiong。“衔接文本与表格数据，实现跨领域文本到 SQL 语义解析”。2020年。arXiv：2012.12627。[125] P. Liu、W. Yuan、J. Fu、Z. Jiang、H. Hayashi 和 G. Neubig。“预训练、提示与预测：自然语言处理提示方法的系统综述”。2021年。arXiv：2107.13586。[126] Y. Liu 等人。“用于神经机器翻译的多语言去噪预训练”。2020年。arXiv：2001.08210。[127] Y. Liu 等人。“RoBERTa：一种鲁棒优化的 BERT 预训练方法”。2019年。arXiv：1907.11692。[128] Y. Liu、S. Pan、M. Jin、C. Zhou、F. Xia 和 P. S. Yu。“图自监督学习：综述”。2021年。arXiv：2103.00111。[129] F. Locatello、S. Bauer、M. Lucic、G. Raetsch、S. Gelly、B. Schölkopf 和 O. Bachem。“对无监督解缠表示学习中常见假设的挑战”。载于：《国际机器学习会议》。PMLR，2019年，第4114–4124页。[130] A. Maas、R. E. Daly、P. T. Pham、D. Huang、A. Y. Ng 和 C. Potts。“用于情感分析的词向量学习”。载于：第49届计算语言学协会年会论文集：人类语言技术。2011年，第142–150页。[131] D. Mahajan 等人。“临床笔记中语义相似句子的识别：采用多任务学习的迭代式中间训练”。载于：《JMIR 医学信息学》8.11（2020年），e22508。[132] S. McCandlish、J. Kaplan、D. Amodei 和 O. D. Team。“大批量训练的经验模型”。2018年。arXiv：1812.06162。[133] A. Merchant、E. Rahimtoroghi、E. Pavlick 和 I. Tenney。“微调期间 BERT 嵌入会发生什么变化？”2020年4月29日。arXiv：2004.14448。[134] S. Merity、C. Xiong、J. Bradbury 和 R. Socher。“指针哨兵混合模型”。2016年。arXiv：1609.07843。[135] T. Mikolov、K. Chen、G. Corrado 和 J. Dean。“向量空间中词表示的高效估计”。2013年。arXiv：1301.3781。[136] T. Mikolov 和 G. Zweig。“上下文相关的循环神经网络语言模型”。载于：2012 IEEE 语音语言技术研讨会（SLT）。IEEE，2012年，第234–239页。[137] G. A. Miller。“WordNet：英语词汇数据库”。载于：《ACM通讯》38.11（1995年），第39–41页。[138] S. Mishra、D. Khashabi、C. Baral 和 H. Hajishirzi。“通过自然语言众包指令实现跨任务泛化”。2022年3月14日。arXiv：2104.08773 [cs]。[139] M. Mitchell。BigScience 大型开放科学、开放获取多语言语言模型。2022年7月6日。URL：https://huggingface.co/bigscience/bloom（访问日期：2022年10月25日）。[140] M. Mosbach、M. Andriushchenko 和 D. Klakow。“关于 BERT 微调的稳定性：误解、解释与强基线”。2021年3月25日。arXiv：2006.04884。[141] A. Mulyar、O. Uzuner 和 B. McInnes。“MT-clinical BERT：通过多任务学习扩展临床信息抽取”。载于：《美国医学信息学会杂志》28.10（2021年），第2108–2115页。[142] S. Narang 等人。“Transformer 修改能否跨实现和应用迁移？”2021年9月10日。arXiv：2102.11972 [cs]。

[143] S. Narayan、S. B. Cohen 和 M. Lapata。“别告诉我细节，只给我摘要！用于极限摘要的主题感知卷积神经网络”。载于：2018年自然语言处理经验方法会议论文集（EMNLP 2018）。比利时布鲁塞尔：计算语言学协会，2018年10月，第1797–1807页。DOI：10.18653/v1/D18-1206。[144] M. Nayyeri、S. Vahdati、C. Aykul 和 J. Lehmann。“采用射影变换的 5* 知识图谱嵌入”。2020年。arXiv：2006.04986。[145] M. Nickel、V. Tresp 和 H.-P. Kriegel。“多关系数据集体学习的三向模型”。载于：国际机器学习会议。2011年。[146] Y. Nie、A. Williams、E. Dinan、M. Bansal、J. Weston 和 D. Kiela。“对抗性自然语言推理：自然语言理解的新基准”。2019年。arXiv：1910.14599。[147] S. J. Nowlan 和 G. E. Hinton。“竞争专家自适应混合的评估”。载于：神经信息处理系统大会。第3卷。1990年，第774–780页。[148] A. van den Oord 等人。“WaveNet：原始音频生成模型”。2016年。arXiv：1609.03499。[149] OpenAI。OpenAI API。2021年。URL：https://beta.openai.com（访问日期：2021年11月14日）。[150] OpenAI。GPT-3 提示示例。2021年9月3日。URL：https://beta.openai.com/examples（访问日期：2021年9月3日）。[151] L. Ouyang 等人。“通过人工反馈训练语言模型遵循指令”。2022年1月31日。arXiv：2203.02155。[152] G. Paass 和 J. Kindermann。“应用于信用评分的重叠叶节点贝叶斯分类树”。载于：《知识发现与数据挖掘研究与发展》。X. Wu、R. Kotagiri 和 K. B. Korb 编。计算机科学讲义。柏林、海德堡：施普林格，1998年，第234–245页。ISBN：978-3-540-69768-8。DOI：10.1007/3-540-64383-4_20。[153] V. Pan。“利用柯西矩阵和多项式进行快速近似计算”。载于：《数学计算》86.308（2017年），第2799–2826页。[154] D. Paperno 等人。“LAMBADA 数据集：需要广泛话语上下文的词语预测”。2016年6月20日。arXiv：1606.06031 [cs]。[155] P. Pasupat 和 P. Liang。“半结构化表格上的组合语义解析”。2015年。arXiv：1508.00305。[156] M. E. Peters、M. Neumann、M. Iyyer、M. Gardner、C. Clark、K. Lee 和 L. Zettlemoyer。“深度上下文化词表示”。载于：北美计算语言学协会人类语言技术分会论文集。2018年，第2227–2237页。[157] M. E. Peters、M. Neumann、R. L. Logan IV、R. Schwartz、V. Joshi、S. Singh 和 N. A. Smith。“知识增强的上下文词表示”。2019年。arXiv：1909.04164。[158] F. Petroni。LAMA：语言模型分析。Meta Research，2020年。URL：https://github.com/ facebookresearch/LAMA（访问日期：2022年3月8日）。[159] F. Petroni、T. Rocktäschel、P. Lewis、A. Bakhtin、Y. Wu、A. H. Miller 和 S. Riedel。“语言模型能否作为知识库？”2019年。arXiv：1909.01066。[160] J. Pfeiffer、I. Vulić、I. Gurevych 和 S. Ruder。“MAD-X：用于多任务跨语言迁移的基于适配器的框架”。2020年。arXiv：2005.00052。[161] J. Pfeiffer 等人。“AdapterHub：适配 Transformer 的框架”。2020年。arXiv：2007.07779。[162] N. Poerner、U. Waltinger 和 H. Schütze。“BERT 尚非知识库：无监督问答中的事实知识与基于名称的推理”。2019年。arXiv：1911.03681。[163] C. Poth、J. Pfeiffer、A. Rücklé 和 I. Gurevych。“预训练什么？高效的中间任务选择”。2021年。arXiv：2104.08247。[164] S. Pradhan、A. Moschitti、N. Xue、O. Uryupina 和 Y. Zhang。“CoNLL-2012 共享任务：在 OntoNotes 中对多语言非受限共指进行建模”。载于：EMNLP-CoNLL 共享任务联合会议。2012年，第1–40页。[165] Y. Pruksachatkun 等人。“面向自然语言理解的预训练模型中间任务迁移学习：何时奏效，为什么奏效？”2020年。arXiv：2005.00628。[166] X. Qiu、T. Sun、Y. Xu、Y. Shao、N. Dai 和 X. Huang。“自然语言处理预训练模型：综述”。载于：《中国科学：技术科学》63.10（2021年6月23日），

第1872–1897页。issn：1674-7321，1869-1900。doi：10 . 1007 / s11431 - 020 - 1647 - 3。arXiv：2003.08271。[167] A. Radford、J. Wu、R. Child、D. Luan、D. Amodei 和 I. Sutskever。“语言模型是无监督多任务学习器”。载于：OpenAI博客 1.8（2019年），第9页。[168] J. W. Rae 等人。“扩展语言模型：训练 Gopher 的方法、分析与洞见”。载于：ArXiv 预印本 ArXiv211211446（2021年12月8日），第118页。[169] J. W. Rae、A. Potapenko、S. M. Jayakumar 和 T. P. Lillicrap。“用于长距离序列建模的压缩 Transformer”。2019年。arXiv：1911.05507。[170] C. Raffel 等人。“利用统一文本到文本转换器探索迁移学习的极限”。载于：机器学习研究杂志 21.140（2020年），第1–67页。[171] c. raffel。C*4* | TensorFlow Datasets。TensorFlow。2019年。网址：https:// www.tensorflow.org/ datasets/catalog/c4（访问于2021年12月14日）。[172] A. Raganato、Y. Scherrer 和 J. Tiedemann。“基于 Transformer 的机器翻译中的固定编码器自注意力模式”。2020年。arXiv：2002.10260。[173] P. Rajpurkar、J. Zhang、K. Lopyrev 和 P. Liang。“Squad：100，000+ 用于文本机器阅读理解的问题”。2016年。arXiv：1606.05250。[174] H. Ren、H. Dai、Z. Dai、M. Yang、J. Leskovec、D. Schuurmans 和 B. Dai。“Combiner：具有稀疏计算成本的全注意力 Transformer”。载于：神经信息处理系统进展 34（2021年）。[175] J. Rodriguez。悟道2.0 的五个关键事实：有史以来构建的最大 Transformer模型。DataSeries。2021年9月21日。网址：https://medium.com/dataseries/five-key-facts-wu-dao-2-0-the-largest-transformer-model-ever-built-19316159796b（访问于2021年12月12日）。[176] A. Rogers、O. Kovaleva 和 A. Rumshisky。“{BERT学}入门：我们对 {BERT} 工作原理的了解”。载于：计算语言学协会会刊 8（2021年），第842–866页。[177] S. Roller、S. Sukhbaatar、A. Szlam 和 J. Weston。“用于大型稀疏模型的哈希层”。2021年。arXiv：2106.04426。[178] A. Romero。GPT-3 吓到你了吗？来认识悟道2.0：拥有1.75万亿参数的巨兽。Medium。2021年6月8日。网址：https://towardsdatascience.com/gpt-3-scared-you-meet-wu-dao-2-0-a-monster-of-1-75-trillion-parameters-832cd83db484（访问于2021年7月29日）。[179] C. Rosset。“Turing-Nlg：微软推出的170亿参数语言模型”。载于：微软博客 — 2020年2月13日（2019年）。[180] A. Roy、M. Saffar、A. Vaswani 和 D. Grangier。“采用路由 Transformer 的高效基于内容的稀疏注意力”。2020年。arXiv：2003.05997。[181] A. Sabeti。GPT-3：一种几乎能写任何东西且好得令人毛骨悚然的人工智能。Arram Sabeti。2020年7月9日。网址：https://arr.am/2020/07/09/gpt-3-an-ai-thats-eerily-good-at-writing-almost-anything/（访问于2021年9月4日）。[182] K. Sakaguchi、R. Le Bras、C. Bhagavatula 和 Y. Choi。“Winogrande：大规模对抗性Winograd模式挑战”。载于：AAAI 人工智能会议论文集。第34卷，第05期。2020年，第8732–8740页。[183] V. Sanh、L. Debut、J. Chaumond 和 T. Wolf。“DistilBERT，BERT 的蒸馏版本：更小、更快、更便宜、更轻量”。2019年。arXiv：1910.01108。[184] T. Schick 和 H. Schütze。“利用完形填空题进行少样本文本分类和自然语言推理”。2021年1月25日。arXiv：2001.07676。[185] T. Schick 和 H. Schütze。“重要的不只是规模：小型语言模型也是少样本学习器”。2021年4月12日。arXiv：2009.07118。[186] J. Schulman、F. Wolski、P. Dhariwal、A. Radford 和 O. Klimov。“近端策略优化算法”。2017年。arXiv：1707.06347。[187] S. Schuster、S. Gupta、R. Shah 和 M. Lewis。“面向多语言任务导向对话的跨语言迁移学习”。2018年。arXiv：1810.13327。[188] J. Sevilla、L. Heim、A. Ho、T. Besiroglu、M. Hobbhahn 和 P. Villalobos。机器学习三个时代的计算趋势。2022年3月9日。doi：10.48550/arXiv.2202.05924。arXiv：2202。05924 [cs]。[189] N. Shazeer。“门控线性单元变体改进 Transformer”。2020年2月12日。arXiv：2002。05202 [cs，stat]。

[190] S. Shen 等人。“Q-BERT：基于海森的 BERT 超低精度量化。”载于：AAAI。2020年，第8815–8821页。[191] T. Shen、Y. Mao、P. He、G. Long、A. Trischler 和 W. Chen。“通过图引导的表示学习挖掘文本中的结构化知识”。2020年。arXiv：2004.14224。[192] T. Shin、Y. Razeghi、R. L. Logan IV、E. Wallace 和 S. Singh。“AutoPrompt：通过自动生成的提示从语言模型中引出知识”。2020年。arXiv：2010.15980。[193] M. Shoeybi、M. Patwary、R. Puri、P. LeGresley、J. Casper 和 B. Catanzaro。“Megatron- Lm：使用模型并行训练数十亿参数语言模型”。载于：arXiv（2019年），arXiv–1909。[194] K. Singla、D. Can 和 S. Narayanan。“学习多语言表示的多任务学习方法”。载于：第56届计算语言学协会年会论文集。第2卷，短文。2018年，第214–220页。[195] D. R. So、W. Mańke、H. Liu、Z. Dai、N. Shazeer 和 Q. V. Le。“Primer：为语言建模搜索高效Transformer”。2022年1月24日。arXiv：2109。08668 [cs]。[196] K. Song、X. Tan、T. Qin、J. Lu 和 T.-Y. Liu。“MASS：用于语言生成的掩码序列到序列预训练”。2019年。arXiv：1905.02450。[197] A. C. Stickland 和 I. Murray。“BERT 及其伙伴：用于多任务学习中高效适配的投影注意力层”。载于：国际机器学习会议。PMLR，2019年，第5986– 5995页。[198] N. Stiennon 等人。“通过人类反馈学习摘要”。载于：神经信息处理系统进展 33（2020年），第3008–3021页。[199] G. Stoica、E. A. Platanios 和 B. Póczos。“Re-Tacred：解决 TACRED 数据集的不足”。载于：AAAI 人工智能会议论文集。第35卷，第15期。2021年，第13843–13850页。[200] F. M. Suchanek、G. Kasneci 和 G. Weikum。“YAGO：语义知识的核心”。载于：第16届万维网国际会议论文集。2007年，第697–706页。[201] P. Sun。发布 SCaNN：高效向量相似性搜索。Google AI 博客。2020 年 7 月 28 日 2020 年 07 。网址：http://ai.googleblog.com/ / /announcing- scann- efficient- vector.html（访问于2021年2月18日）。[202] T. Sun、Y. Shao、X. Qiu、Q. Guo、Y. Hu、X. Huang 和 Z. Zhang。“CoLAKE：上下文化语言与知识嵌入”。2020年。arXiv：2010.00309。[203] Y. Sun 等人。“ERNIE：通过知识融合增强表示”。2019年。arXiv：1904.09223。[204] Z. Sun、H. Yu、X. Song、R. Liu、Y. Yang 和 D. Zhou。“MobileBERT：面向资源受限设备的紧凑型任务无关BERT”。2020年4月14日。arXiv：2004.02984。[205] N. Tang 等人。“RPT：关系型预训练Transformer几乎是实现数据准备普及化所需的一切”。2020年。arXiv：2012.02469。[206] Y. Tay、D. Bahri、D. Metzler、D.-C. Juan、Z. Zhao 和 C. Zheng。“Synthesizer：重新思考 Transformer模型中的自注意力”。2021年5月24日。arXiv：2005。00743 [cs]。[207] Y. Tay、M. Dehghani、D. Bahri 和 D. Metzler。“高效Transformer：综述”。2020年。arXiv：2009.06732。[208] Y. Tay、Z. Zhao、D. Bahri、D. Metzler 和 D.-C. Juan。“HyperGrid Transformer：迈向适用于多项任务的单一模型”。载于：国际学习表示会议。2021年。[209] Y. Tay 等人。“Long Range Arena：高效Transformer的基准测试”。2020年。arXiv：2011.04006。[210] N. Tripuraneni、M. Jordan 和 C. Jin。“论迁移学习理论：任务多样性的重要性”。载于：神经信息处理系统进展 33（2020年），第7852– 7862页。[211] L. TriviaQA。CodaLab - 竞赛。2021年2月28日。网址：https://competitions.codalab.org/competitions/17208#results（访问于2021年2月28日）。[212] A. Vaswani 等人。“Attention Is All You Need”。载于：神经信息处理系统进展。2017年，第5998–6008页。[213] P. Verga、H. Sun、L. B. Soares 和 W. W. Cohen。“事实即专家：基于符号知识的可适配、可解释神经记忆”。2020年。arXiv：2007.00849。

[214] D. Vrandečić 和 M. Krötzsch。《维基数据：一个自由协作知识库》。载于：《ACM通讯》57.10（2014），第78–85页。[215] K. Wali。EleutherAI 推出 GPT-NeoX-20B：最大的公共访问语言模型。《Analytics India Magazine》。2022年2月14日。网址：https://analyticsindiamag.com/eleutherai-launches-gpt-neox-20b-the-biggest-public-access-language-model/（访问于 2022年2月23日）。[216] J. Wallat、J. Singh 和 A. Anand。《BERTnesia：探究 BERT 中知识的获取与遗忘》。2020年。arXiv：2010.09313。[217] A. Wang、A. Singh、J. Michael、F. Hill、O. Levy 和 S. R. Bowman。《GLUE：自然语言理解的多任务基准与分析平台》。2018年。arXiv：1804.07461。[218] A. Wang、A. Singh、J. Michael、F. Hill、O. Levy 和 S. R. Bowman。《Glue：自然语言理解的多任务基准与分析平台》。2019年2月22日。arXiv：1804.07461。[219] A. Wang 等人。《SuperGLUE：一个更具挑战性的通用语言理解系统基准》。载于：神经信息处理系统进展，2019年，第3266–3280页。[220] B. Wang。EleutherAI - 文本生成测试界面。2021年。网址：https://6b.eleuther.ai/（访问于 2021年11月14日）。[221] B. Wang。Mesh-Transformer-JAX：使用 JAX 实现的 Transformer 语言模型模型并行方案。2021年5月1日。网址：https://github.com/kingoflolz/mesh-transformer-jax（访问于 2021 年 11 月 14 日）。[222] R. Wang 等人。《 K-Adapter：通过适配器将知识注入预训练模型》。2020年12月28日。arXiv：2002.01808。[223] W. Wang 等人。《StructBERT：将语言结构融入深层语言理解预训练》。2019年。arXiv：1908.04577。[224] X. Wang、T. Gao、Z. Zhu、Z. Liu、J. Li 和 J. Tang。《KEPLER：知识嵌入与预训练语言表示的统一模型》。2020年11月23日。arXiv：1911.06136。[225] Z. Wang、A. W. Yu、O. Firat 和 Y. Cao。《迈向零标签语言学习》。2021年9月19日。arXiv：2109。09193 [cs]。[226] J. Wei、X. Wang、D. Schuurmans、M. Bosma、E. Chi、Q. Le 和 D. Zhou。《思维链提示能激发大语言模型的推理能力》。2022年。arXiv：2201.11903。[227] J. Wei 等人。《微调语言模型就是零样本学习器》。载于：国际学习表征会议 2022（2022），第46页。[228] X. Wei、Y. Hu、R. Weng、L. Xing、H. Yu 和 W. Luo。《关于跨语言学习通用表示》。2020 年。arXiv：2007.15960。[229] A. Williams、N. Nangia 和 S. R. Bowman。《通过推理进行句子理解的广覆盖挑战语料库》。2017年。arXiv：1704.05426。[230] G. Wilson 和 D. J. Cook。《无监督深度领域适应综述》。载于：《ACM智能系统与技术汇刊》TIST 11.5（2020），第1–46页。[231] G. I. Winata、A. Madotto、Z. Lin、R. Liu、J. Yosinski 和 P. Fung。《语言模型是少样本多语言学习器》。2021年9月15日。arXiv：2109.07684。[232] S. Wu 和 M. Dredze。《Beto、Bentz、Becas：BERT 令人惊讶的跨语言有效性》。载于：2019年自然语言处理实证方法会议论文集、第9届自然语言处理国际联合会议（EMNLP-IJCNLP）。EMNLP-IJCNLP 2019。中国香港：计算语言学协会，2019年11月，第833–844页。doi：10.18653/v1/D19-1077。[233] J. Xia、Y. Zhu、Y. Du 和 S. Z. Li。《图上预训练综述：分类体系、方法与应用》。2022年。arXiv：2202.07893。[234] W. Xiong、J. Du、W. Y. Wang 和 V. Stoyanov。《预训练百科：弱监督知识预训练语言模型》。2019年。arXiv：1912.09637。[235] L. Xue。mT5-code：多语言T5。Google Research，2021年2月25日。网址：https://github.com/google-research/multilingual-t5（访问于 2021年2月26日）。[236] L. Xue 等人。《mT5：大规模多语言预训练文本到文本 Transformer》。2020年。arXiv：2010.11934。

[237] I. Yamada、A. Asai、H. Shindo、H. Takeda 和 Y. Matsumoto。《LUKE：通过实体感知自注意力实现深度上下文化实体表示》。2020年。arXiv：2010.01057。[238] J. Yang 等人。《GraphFormers：用于文本图表示学习的 GNN 嵌套 Transformer》。载于：神经信息处理系统进展 34（2021）。[239] Z. Yang、Z. Dai、R. Salakhutdinov 和 W. W. Cohen。《突破 Softmax 瓶颈：高秩 RNN 语言模型》。2017年。arXiv：1711.03953。[240] Z. Yang、Z. Dai、Y. Yang、J. Carbonell、R. R. Salakhutdinov 和 Q. V. Le。《XLNet：用于语言理解的广义自回归预训练》。载于：神经信息处理系统进展，2019年，第5753–5763页。[241] P. Yin、G. Neubig、W.-t. Yih 和 S. Riedel。《TaBERT：文本和表格数据的联合理解预训练》。2020年。arXiv：2005.08314。[242] W. Yin。《少样本自然语言处理元学习综述》。2020年。arXiv：2007.09604。[243] W. Yu、M. Jiang、Z. Hu、Q. Wang、H. Ji 和 N. Rajani。《知识增强自然语言生成》。载于：（2021年11月10日），第6页。[244] W. Yu、C. Zhu、Z. Li、Z. Hu、Q. Wang、H. Ji 和 M. Jiang。《知识增强文本生成综述》。2021年7月5日。arXiv：2010.04389。[245] W. Yuan、G. Neubig 和 P. Liu。《BARTScore：将生成文本作为文本生成任务进行评估》。载于：神经信息处理系统进展 34（2021）。[246] C. Yun、Y.-W. Chang、S. Bhojanapalli、A. S. Rawat、S. J. Reddi 和 S. Kumar。《$ O (n) $ 个连接已足够强大：稀疏 Transformer 的通用逼近能力》。2020年。arXiv：2006.04862。[247] M. Zaheer 等人。《Big Bird：面向更长序列的 Transformer》。载于：神经信息处理系统进展 33（2021年1月8日）。[248] W. Zeng 等人。《PanGu-$$\backslash$alpha $：采用自动并行计算的大规模自回归预训练中文语言模型》。2021年。arXiv：2104.12369。[249] B. Zhang 和 R. Sennrich。《均方根层归一化》。2019年。arXiv：1910.07467。[250] J. Zhang、H. Zhang、C. Xia 和 L. Sun。《Graph-BERT：学习图表示只需注意力》。2020年1月22日。arXiv：2001。05140 [cs, stat]。[251] J. Zhang、Y. Zhao、M. Saleh 和 P. Liu。《PEGASUS：通过提取间隙句进行抽象式摘要预训练》。载于：国际机器学习会议。PMLR，2020年，第11328–11339页。[252] L. Zhang。《迁移适应学习：十年综述》。2019年。arXiv：1903.04687。[253] S. Zhang 等人。OPT：开放式预训练Transformer语言模型。2022年5月5日。arXiv：2205。01068 [cs]。[254] Y. Zhang、V. Zhong、D. Chen、G. Angeli 和 C. D. Manning。《位置感知注意力和监督数据提升槽位填充效果》。载于：2017年自然语言处理实证方法会议论文集。2017年，第35–45页。[255] Z. Zhang、X. Han、Z. Liu、X. Jiang、M. Sun 和 Q. Liu。《ERNIE：融合信息实体的增强语言表示》。2019年6月4日。arXiv：1905.07129。[256] Z. Zhang、F. Qi、Z. Liu、Q. Liu 和 M. Sun。《了解哪些是不需要的：针对注意力头的单次元剪枝》。载于：AI Open 2（2021），第36–42页。[257] A. Zhavoronkov。悟道 2 19, 2021 0 2021 07 19 20 .——来自中国的更大、更强、更快的人工智能。《福布斯》。7月。网址：https://www.forbes.com/sites/alexzhavoronkov/ / / / wu-dao-bigger-stronger-faster-ai-from-china/（访问于 2021年7月29日）。[258] C. Zhu、W. Ping、C. Xiao、M. Shoeybi、T. Goldstein、A. Anandkumar 和 B. Catanzaro。《长短 Transformer：用于语言与视觉的高效 Transformer 》。载于：神经信息处理系统进展 34（2021）。[259] F. Zhu、W. Lei、C. Wang、J. Zheng、S. Poria 和 T.-S. Chua。《检索与阅读：开放域问答综合综述》。2021年。arXiv：2101.00774。[260] F. Zhuang 等人。《迁移学习综合综述》。载于：IEEE 论文集 109.1（2020），第43–76页。[261] B. Zoph 等人。《设计高效稀疏专家模型》。2022年。arXiv：2202.08906。

第4章
基础模型获取的知识

摘要 在预训练期间，基础模型在大量文档集合上进行训练，并学习正确、流畅语言中的词分布。本章将研究预训练语言模型和更大型基础模型所获取的知识。我们首先讨论将基础模型应用于特定基准测试，以检验其在大量领域中的知识，并考察模型是否能够根据内容得出正确结论。另一组测试通过补全文本，以及应用分别考察句法知识、语义知识和逻辑推理的特定探测分类器，来评估基础模型。最后，我们将研究这些基准测试是否可靠且具有可复现性，即其他研究人员重复进行测试时，它们是否确实检验了目标性质，并得到相同的性能值。

关键词： 基础模型中的知识、常识知识、逻辑连贯性、基准集合、可复现性

在预训练期间，预训练语言模型（PLM）和更大型基础模型在大量文档集合上进行训练，并学习正确、流畅语言中的词分布。在微调期间，模型利用预训练所获得的知识适应特定任务，而只需少量人工标注的微调数据。本章将通过不同类型的测试研究这些模型所获取的知识：

- 首先，我们通过特定的基准测试评估预训练语言模型和基础模型，以检验它们在众多领域中的知识，并考察模型能否根据内容得出正确结论（第 4.1 节）。通常，这些基准集合会汇总不同测试的结果，给出综合性能指标。基准测试可以通过微调模型来执行特定的分类任务，也可以通过少样本查询基础模型来完成。

• 接着，我们通过补全文本并应用特定的探测分类器来评估基础模型，而不调整模型参数（第 4.2 节）。我们分别考察句法知识、语义知识和逻辑推理，并展示不同领域以及不同模型架构的成就与不足。

• 最后，我们将考察基准测试是否可靠，即是否确实测试了目标性质（第 4.3 节）。此外，我们还会分析已发表的基准测试结果能否复现，以及由其他研究人员重复测试时，是否会得到相同的性能值。

4.1 基准集合

为了对常识知识和常识推理进行量化，学界编制了多个基准测试。这些基准测试支持对自然语言理解的不同方面进行标准化比较，并提供可比较的得分，以评估不同预训练语言模型的优势和不足。基准测试一直是推动语言模型发展的关键因素。PapersWithCode [45] 提供了全面的基准测试集合及相应的排行榜。Storks 等人 [62] 对实际使用的基准测试进行了综述。

要公平比较模型架构，就必须确保参数数量、训练数据规模和训练所需的计算开销相近。第 3.5.1 节对此进行了充分讨论。因此，许多作者会开展大量消融研究，将训练资源调整到一个标准，例如以 BERT 作为“基准模型”。这非常重要，因为它有助于读者直观了解预训练资源的影响。尽管如此，两方面的问题仍常常妨碍可比性：

1. 一些训练数据集（例如 BERT 的 BooksCorpus）并未公开提供。2. 这些比较并未展示当数据规模、参数数量或计算开销增加时模型的性能。

因此，像“模型架构 A 在执行任务 X 方面优于模型架构 B。”这样的陈述通常并不成立，而必须加以限定 [2], ，例如：“当使用来自领域 Y 的低/高质量数据语料库进行预训练，并投入较少/较多计算资源 Z 时，模型架构 A 在执行任务 X 方面优于模型架构 B 。”

表 4.1 四种不同模型和人工标注员在 GLUE 基准上的结果。每项任务中，预训练语言模型的最佳值以粗体标出 [18, p. 7]。优于所有模型得分的人工得分用下划线标出。

模型	CoLA	QQP	MNLI m	SST-2	STS-B	QNLI	RTE	WNLI	MRPC	平均值
	Mcc	Acc	Acc	Acc	Corr	Acc	Acc	Acc	F1	
	语法改写		蕴含与情感		相似问题		蕴含	共指	释义	
人工 [42]	66.4	80.4	92.0	97.8	92.7	91.2	93.6	95.9	86.3	87.1
$BERT_{LARGE}$	60.6	91.3	86.6	93.2	90.0	92.3	70.4	65.1	88.0	84.1
$RoBERTa_{LARGE}$	68.0	92.2	90.2	96.4	92.4	93.9	86.6	89.9	90.9	88.8
$XLNET_{LARGE}$	69.0	92.3	90.8	**97.0**	92.5	94.9	85.9	92.5	90.8	89.2
$DeBERTaV3_{LARGE}$	**75.3**	**93.0**	**91.8**	96.9	**93.0**	**96.0**	92.7	-	**92.2**	**91.4**

4.1.1 GLUE 基准集合

为了测试预训练语言模型理解文档内容的能力，人们开发了 GLUE（第 2.1.5 节）基准测试套件。这是一组包含 9 个基准测试的集合，用于测试自然语言理解（*NLU*）的不同方面。综合性能通过单个得分进行衡量，人工标注者的得分为 87.1。第 2.1 节表格 2.1 通过示例详细描述了这些任务。事实证明，针对不同 GLUE 任务进行微调的 BERT 变体可以取得比人类更好的结果。这些结果针对模型的大型变体确定，并显示在表格 4.1 中。

过去几年里，GLUE 常被用于展示预训练语言模型的自然语言理解能力。目前，DeBERTaV3 [18] （第 3.1.1 节）经微调后的平均分最高，达到 91.4。它为内容和位置分别使用嵌入，并采用相应的解耦注意力机制。预训练语言模型只有在三项任务上的表现不如人类，但差距很小。请注意，由多个模型组成的集成模型通常会取得略好的结果。Nangia等人 [42] 还测量了由 5 人组成的人类团队的表现。这些数字无法比较，因为当团队成员的判断不一致时，相关案例被排除在外。PaLM 等较新的模型使用 SuperGLUE 而非 GLUE，因为 GLUE 被认为过于简单。

4.1.2 SuperGLUE：GLUE 的高级版本

由于过去几年的进步，预训练语言模型在大多数任务上的表现已达到人类水平，而 GLUE 已无法再区分不同模型。因此，GLUE 的作者提出了一个要求更高的测试套件，称为**SuperGLUE** [68]，作为

表格 4.2 人工标注员和五种不同模型在 SuperGLUE 基准测试测试集上的结果。每项任务的最佳值以粗体显示，优于模型值的人工标注结果以<u>下划线</u>标示。GPT-3 的少样本结果报告为 FS 值，其他结果则为微调后的值。

模型	BoolQ	CB		COPA	MultiRC		ReCoRD		RTE	WiC	WNLI	平均值
	Acc	准确率/F1值		准确率	F1值a/精确匹配		F1值/精确匹配	F1值/精确匹配	Acc	Acc	Acc	
	问答 是/否	蕴含		原因	问答乘法		实体	蕴含	WSD	共指		
人工 [68]	89.0	95.8/98.9		<u>100.0</u>	81.8/51.9		91.7/91.3	93.6	<u>80.0</u>	100.0	89.8	
$BERT_{336M}$ [68]	77.4	83.6/75.7		70.6	70.0/24.0		72.0/71.3	71.6	69.5	64.3	69.0	
$GPT\text{-}3_{270B}$ FS [7]	76.4	75.6/52.0		92.0	75.4/30.5		91.1/90.2	69.0	49.4	80.1	71.8	
DeBERTA 集成 [19]	90.4	94.9/97.2		98.4	88.2/63.7		94.5/94.1	93.2	77.5	95.9	90.3	
$PaLM_{540B}$ [9]	91.9	94.4/96.0		99.0	88.7/63.6		94.2/93.3	**95.9**	77.4	95.9	90.4	
$ST\text{-}MoE_{32B}$ [73]	**92.4**	**96.9/98.0**		**99.2**	**89.6/65.8**		**95.1/94.4**	93.5	**77.7**	**96.6**	**91.2**	

GLUE 的高级版本，包含八项高难度任务。这些任务与 GLUE 中的任务相似，但需要考虑更长的上下文。

- *BoolQ*是一项问答任务，问题采集自谷歌搜索，答案为是/否。

- *CB* 是一项文本蕴含任务。
- *COPA* 是一项因果推理任务，系统必须从两个可能的选项中确定给定前提的原因或结果。
- *MultiRC* 是一项问答任务，每个实例由一段上下文段落、一个关于该段落的问题以及一系列可能的答案组成。
- 在*ReCoRD* 中，每个示例由一篇新闻文章以及其中一个实体被掩码的文章组成。系统必须从一系列可能的实体中预测被掩码实体。

- *RTE* 要求检测一个假设是否可由一个前提蕴含。
- *WiC* 是一项词义消歧任务，系统需要判断在给定的两个句子中，多义词在两个句子里的词义是否相同。

- *WSC* 是 Winograd模式挑战，系统需要确定代词所指代的正确名词短语。

性能仍以一个平均得分衡量，人工标注员的值为 89.8 [66]。

GPT-3 [7] 是一种巨大的语言模型（第3.1.2节），无需微调即可通过指令执行任务（第3.2节）。借助这种少样本学习，GPT-3的平均SuperGLUE得分仅为71.8，如表4.2所示。显然，对特定任务进行微调似乎十分重要。最近，经过微调的DeBERTa集成模型（第3.1.1节）在SuperGLUE上超越了人类表现，平均得分达到90.3。最困难的任务是比较两个句子中的词义（WiC），其准确率约可达到77%。自回归语言模型*PaLM* 540B在SuperGLUE上经过微调后，在

测试集 [9, p. 13]上取得了90.4%的平均得分。最佳平均得分为91.2%，由$ST\text{-}MoE_{32B}$混合专家模型（第3.5.2节）取得，该模型拥有269B个参数 [73]。这表明基础模型能够分析复杂的文本语义。

GLUE 和 SuperGLUE 受到过批评，因为所提出问题的答案总是可以归结为分类任务，而且系统不必用自然语言表述答案。此外，事实证明，预训练语言模型的性能并不十分稳定。研究表明，如果替换某些词语，当前模型的预测往往会以不一致的方式发生变化 [51]。例如，在情感分析中，如果输入“我喜欢这趟航班”被分类为正面，那么“我不喜欢这趟航班”就不应被分类为中性。Ribeiro 等人 [51] 表明，此类不一致现象时常发生。他们开发了**CheckList** 系统（第 4.3.1 节），用于自动生成测试样例，以探测模型的性能。

4.1.3 文本补全基准

自回归语言模型的任务是可靠地生成文本中的下一个单词。这既要符合语法正确性，也要保证语义一致性。*LAMBADA* 基准测试 [44]是展示这项能力的良好测试。它包含来自 BooksCorpus 的约 10,000 个段落，这些段落摘自未出版的小说。任务是预测每个段落最后一句中缺失的最后一个单词。人工筛选示例，以确保模型必须考虑至少包含 50 个词元的完整段落，才能推断出最后一个词。

一个示例段落是“树荫下洒落着点点阳光，树下凉爽的草地也令人舒适；在闷热的厨房里待过之后，这里是个令人欣喜的歇脚处。我很高兴能倚着树那粗糙易碎的树皮休息，并开始享用我的早餐：涂着黄油的烤面包和新鲜水果。甚至连水都很好喝，因为它如此干净、冰凉。这几乎弥补了缺少的东西。”，其中“咖啡”是需要预测的缺失目标词。省略了~~那些~~较简单的语言模型也能轻易预测的示例。只有当人工能够根据完整段落预测目标词、却无法仅根据最后一句预测时，才会选取该示例。

GPT-3_{175B} 自回归语言模型 [48] 以 76.2% 的 [7, p. 12]预测出了最后一个单词。PaLM_{540B} 通过少样本指令将准确率提高到了 89.7 [9, p. 79]。这意味着，在测试数据中，预测出的单词有近十分之九恰好就是缺失的单词。

另一个与语言建模相关的基准测试是由维基百科 2.8 万篇文章组成的*WikiText-103* [38]，包含 1.03 亿个词元。如果将大型基础模型应用于该语料库，可得到以下困惑度：$\text{GPT-2}_{1.7B}$ 17.5 [48],Megatron-LM 为 10.8 [58],Gopher 为 $_{280B}$ 8.1 [49, p. 61]。最近，一个小型 $\text{Retro}_{1.8B}$ 模型借助检索将困惑度降至 3.9 [49, p. 12]。请注意，WikiText 103 与 Retro 的训练数据可能存在部分重叠，而去重并未检测到这一点。

表格 4.3 Gopher 及相关模型的评估基准测试分组 [49, 页。 8]。

任务组	# 任务	示例
语言建模	20	WikiText-103、The Pile：PG-19、arXiv、FreeLaw、.....
阅读理解	3	RACE-m、RACE-h、Lambada
事实核查	3	FEVER（双向与三向）、MultiFC
问答	3	Natural Questions、TriviaQA、TruthfulQA
常识	4	HellaSwag、Winogrande、PIQA、SIQA
大规模多任务语言理解（MMLU） [20]	57	高中化学、天文学、临床知识、社会科学、数学，.....
BIG-bench [60]	62	因果判断、认识论推理、时间序列、逻辑、数学、代码、社会推理，.....

4.1.4 大型基准集合

最近，人们开发了 GPT-3、Gopher 和 PaLM 等大型自回归语言模型，它们使用包含数千亿个词元的超大型文档集合进行训练。这些模型应能在广泛的任务中表现出色。因此，评估其性能时，不再局限于 GLUE 基准，而是采用大量涵盖各种可能应用方面的基准测试。

一种常见观点认为，当前的基准测试并不充分，而且已经“达到饱和”“存在缺陷”，并且“被研究人员过度拟合”。Bowman 等人 [5] 认为，“许多自然语言理解（NLU）任务的评估已经失效”。他们指出，排行榜上的一些顶尖系统连简单的测试用例都无法通过（参见 [51]）。因此，他们提出了新基准测试的四项要求：

- 只有当模型在实际应用中也表现出色时，它在基准测试中才应取得良好性能。
- 基准测试的标注应准确且无歧义（例如，Natural Questions 中有 36% 的答案存在歧义）。
- 基准测试应足够大且具有挑战性，以便发现模型之间相关的性能差异。

基准测试应揭示系统中可能造成危害的社会偏见，同时不应助长偏见的产生。

他们总结了一些有望应对这些挑战的进展，包括由众包工作者和领域专家共同参与的数据收集，以及规模更大的数据验证。

为回应这一批评，人们定义了两个全面的基准测试集合。大规模多任务语言理解（MMLU）基准测试 [20]通过包含选择题的人类考试来进行模拟，每道题有四个回答。除了逻辑和数学推理之外，它还测试模型在广泛学术学科中的能力，涵盖从计算机科学到历史和法律。另一个

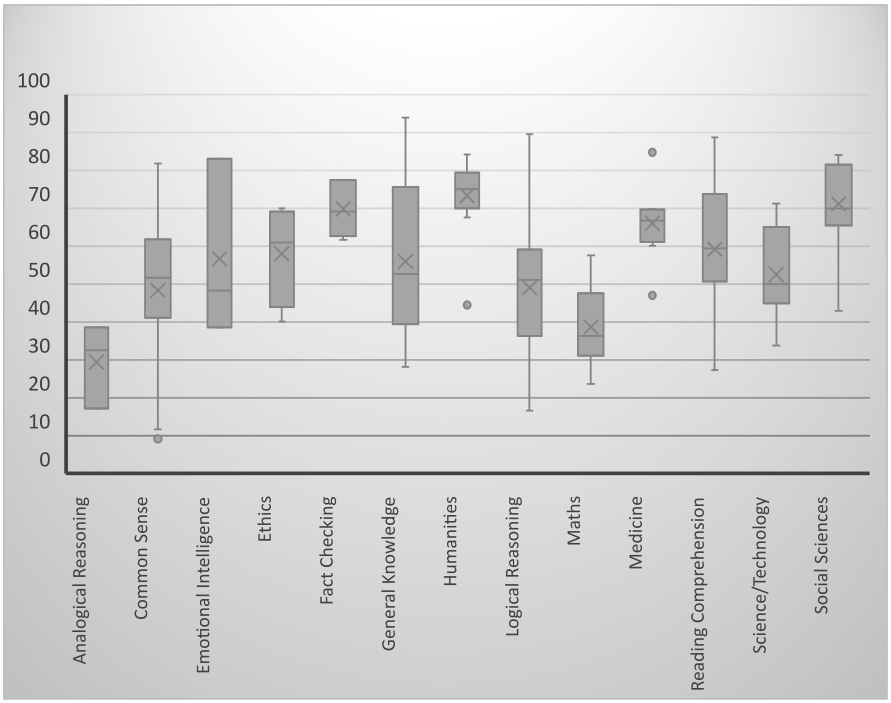


图 4.1 不同组在 152 项不同基准测试上的准确率（百分比），这些基准测试用于评估 Gopher 模型 [49, 页。 57]。箱体表示第 25 和第 75 百分位数，内部线表示中位数。外侧线表示上下四分位数之外的变异范围。

集合是*BIG-bench* 协作基准测试 [1, 60]„ 旨在评估阅读理解、问答、世界理解等语言理解方面。两个基准测试集合都包含一百多个任务。

参数量为 280B 的 *Gopher* 模型，以及 GPT-3、Jurassic-1 和 Megatron-Turing NLG 等其他模型（均在第 3.1.2 节讨论过），都在这些及其他基准测试上接受了测试。请注意，这些测试总共使用了表 4.3 所述的 152 项基准测试。Gopher 在 124 项任务中的 100 项（81%）上取得了改进，超过先前最先进水平的得分。在语言建模（下一个词预测）方面，Gopher 在 19 项基准测试中的 10 项上提升了最先进水平。请注意，所有基准测试结果都不是通过微调获得的，而是通过零样本或少样本学习获得的。

图 4.1 展示了 Gopher 在各主题组中的准确率分布。Gopher 在 7 项数学任务中的 4 项、9 项常识任务中的 5 项、12 项逻辑推理任务中的 9 项、24 项事实核查和一般知识任务中的 22 项、全部 24 项 STEM（科学、技术、工程和数学）及医学任务、全部 15 项人文学科和伦理学任务，以及 11 项阅读理解任务中的 10 项，均提升了最先进水平。常识和一般知识任务的平均准确率约为

50%，这表明模型具备一定的相关知识，但仍有提升空间。这些测试中包括逻辑推理基准测试，例如“形式谬误、三段论与否定”或“逻辑谬误检测”。19 项基准测试中，准确率超过 60% 的只有两项 [49,页。 58], 这表明，即使对于如此大型的模型，正确推理仍是一项重大挑战。显然，这一系列评估深入展现了所比较模型的能力。可以预期，在语言生成过程中执行检索的新模型 Retro（第 6.2.3 节）将改善这些结果。

近期，参数量为 580B 的 *PaLM* 自回归语言模型 [9,p. 15] 接受了 BIG-bench 基准的评估。在这 150 项任务中，PaLM 使用 5-shot提示取得了 46% 的归一化平均得分，高于人工平均得分 39%。然而，最佳人类专家的得分约为 77%。不同 BIG 基准测试领域的详细结果尚未公布。在 58 项 BIG任务的子集中（这些任务也曾用于评估先前的模型），PaLM 的 5-shot归一化得分约为 55%，再次高于人工平均得分 49%，并优于 Chinchilla（47%）和 Gopher（30%）。GPT-3 在这 58 项任务上的 1-shot性能为 16%。总体而言，像 Gopher 和 PaLM 这样拥有数千亿参数的基础模型，在 BIG 基准测试上的结果“显著更好”，即便其模型架构并无根本差异 [1]。从这一点来看，基础模型展现出一种定性上全新的行为。

Google 的研究人员提出，使用目前包含 200 项任务的 BIG-bench 基准来替代图灵测试，以检验“智能” [61]。这样便可大规模评估人工智能系统的知识。最近，一位 Google 工程师发表了与 LaMDA 语言模型的对话 [31] （第 6.6.3 节）。在他看来，这表明 LaMDA 具有“感知能力”。然而，BIG 等知识与推理测试并不检验人类智能的这一方面，因此需要开发新类型的测试。

4.1.5 总结

基准集合是展示预训练语言模型在特定任务上优越性的常用方式。不过，要展现某种架构的优点，还必须报告并比较参数数量、训练数据规模和计算开销，因为这些数值也会影响模型性能。

由九项语言理解任务组成的GLUE基准集合展现了预训练语言模型在过去几年中的显著进步。它测试预训练语言模型检测释义、共指关系、逻辑蕴涵和语法正确性的能力。同时，平均准确率超过了平均人类表现。类似但更具挑战性的 SuperGLUE 基准套件也已推出，其中人类表现同样被超越。对于自回归语言模型而言，LAMBADA 基准测试要求模型具备令人印象深刻的能力，以确定段落中最可能出现的最后一个单词。当前的模型（如PaLM）能够

以接近90%的准确率预测最后一个单词，展现出其捕捉论证脉络的能力。

基础模型通常通过涵盖诸多方面的广泛标准化测试集进行测试，例如常识知识、情商、逻辑推理或社会科学。近期的基础模型，如 Gopher 和 PaLM，拥有数千亿参数，其性能已经超过人类平均水平，并且“显著优于”规模更小的模型。然而，这些模型的准确率仍低于人类专家。尽管这些基准测试具有很强的表现力，但它们没有考虑模型的社会影响，也无法检测自我意识和感知能力等特征。

4.2 通过探测分类器评估知识

本节考察预训练语言模型在多大程度上习得不同类型的知识。我们讨论小型BERT模型所涵盖的知识，随后评述 GPT-3 和 PaLM 等基础模型的改进。首先，我们考察它们对正确语言的句法知识。然后，我们研究预训练语言模型表征了多少常识知识。最后，我们探讨预训练语言模型生成的输出是否在逻辑上保持一致。

4.2.1 BERT 的句法知识

我们以 BERT 为例，讨论预训练语言模型中所蕴含的句法知识。在预训练过程中，BERT 能够捕捉句法知识 [54]。嵌入可以编码有关词类、句法短语和句法角色的信息。探测分类器可以以 85% 的准确率预测词性标注和上位语义类别信息 [33]。显然，这些信息必须编码在 BERT 的最终嵌入中。BERT 还掌握主谓一致 [17] 和语义角色方面的知识 [14]。此外，还可以从 BERT 中提取依存树和句法成分树 [21, 23, 27]。虽然探测表明可以从表示中提取这些信息，但也可以证明[13]，在某些情况下，这些特征并未用于预测。根据一项实证评估，预训练语言模型在底层用短语特征编码语言学信息，在中间层编码句法特征，在顶层编码语义特征 [23]。

然而，BERT 的句法知识并不完整，例如，有证据表明 BERT 经常无法捕捉否定。例如，在掩码句子“一只知更鸟是一只 [掩码]” 中，$\text{BERT}_{\text{LARGE}}$能够以较高的概率 [14]确定正确的上位义，例如“鸟”。另一方面，对于“一只知更鸟不是一只 [掩码]”中的掩码，模型以最大概率预测“知更鸟”、“鸟”、“企鹅”、“男人”、“飞”，实际上忽略了否定。

第 4.1 节中介绍的一些基准测试会检验预训练语言模型的句法知识。一个示例是 GLUE 的 CoLA 任务，该任务测试句子的语法正确性，是 GLUE 中最困难的任务，最佳模型只能获得约 75% 的正确答案（表 4.1）。SuperGLUE（第 4.1.2 节）是一项需要句法知识的基准测试，例如文本蕴含任务 COPA 和共指消解任务 WSC。微调后的 BERT 平均得分为 69.0，而微调后的 PaLM_{540B} 达到 91.4 的平均得分（表 4.2）。PaLM 等大型基础模型的参数量是 BERT 的 1,000 多倍，显然能够比“较小的”BERT 更好地捕捉句法知识。

4.2.2 常识知识

世界知识，也称为常识知识，由关于我们日常世界的事实构成，例如“火是热的”。检验世界知识的一种简单方法是用完形填空陈述查询 BERT，例如“爱因斯坦出生于[掩码]”。BERT 获得了一些关于语义角色的语义知识，并对实体类型和关系的信息进行编码 [54]。例如，在句子“给一位 [掩码]” 中，词元“服务员”在[掩码]的位置上具有较高的概率。Petroni 等人 [46] 和 Zhou 等人 [72] 对这类查询进行了实验，并得出结论：BERT 包含的世界知识，与传统的监督式信息抽取方法相比具有竞争力。研究表明，BERT 的上下文嵌入会形成与词义相对应的簇 [56]。这解释了为什么 BERT 很擅长词义消歧（图 2.10）。

Petroni 等人 [46] 指出，标准语言模型预训练方法更容易学会某些类型的事实知识，而其他类型则难得多。他们表示，无需微调，BERT 所包含的关系知识就足以与能够获取一些预言机知识的传统自然语言处理方法相媲美。此外，在开放域问答方面，BERT 的表现也显著优于监督基线。BERT 的这些能力是一项了不起的成就。

语言模型 GPT-3 的参数数量是 BERT 的一百倍，其常识知识也出色得多。例如，可以从它对以下问题（Q）的回答（A）中看出这一点：*“Q*：有没有三条腿的动物？”“*A*：没有，没有三条腿的动物。”或“*Q*：足球运动员和汽车哪个更重？”“*A*：汽车比足球运动员重。” [29]。在一项初步实验中，研究人员请八十位参与者判断篇幅为 200 个单词的短文是由人类还是 GPT-3 撰写的。参与者有 48% 的时候判断错误，表现仅略好于随机猜测 [7]。

然而，预训练语言模型的语义知识并不完美。例如，BERT 在数字表示方面存在困难，而且常常难以替换命名实体 （命名实体s），例如人名或地名。例如，在共指任务中替换姓名，会改变 85% 的指向同一实体的表达所对应的共指指派 [3]。显然，预

训练版 BERT 难以将涉及某个命名实体的关系泛化到同一类型的其他命名实体。此外，BERT 难以转移基于物体角色或类型的知识。另外，只需在完形填空查询中添加一些内容，就可能误导 BERT。一个例子是“*Talk*”出现在“*Talk*? *Birds can* [*MASK*]”中。人类会忽略“*Talk*?”，并运用世界知识生成类似“*fly*”这样的结果。相比之下，预训练语言模型可能会受到误导，为掩码给出错误答案“*talk*” [26]。

另一个相关现象是对释义的不变性。Elazar 等人 [12] 生成了一组高质量的释义，共 328 条，用以表达 38 种关系。例如，“*X* 最初在 [掩码]播出”和“*X* 于 [掩码]首播”，如果将 [掩码], 中的“*X*”替换为《宋飞正传》等电视剧，模型就应该给出相同的预测。尽管在约 60% 的情况下，模型能够获取正确填充掩码所需的知识，但 $\text{BERT}_{\text{LARGE}}$ 只有在 48.7% 的情况下能对释义保持一致。这表明，训练数据中出现的事实并非都会编码到参数中，而且模型也并非总能识别释义等价性。模型变体 RoBERTa 和 ALBERT 的一致性更低，尽管它们在其他任务上的表现优于 BERT。

考察词序对 BERT 性能的影响颇有启发。词序由特定的位置嵌入体现，这些嵌入会添加到词元嵌入中。然而，事实证明，即使打乱词元位置，BERT 等掩码语言模型仍能达到较高的准确率。为了进行预训练，Sinha 等人 [59]执行了句子置换，即随机将句子中的每个单词放到不同的位置。随后，模型在 GLUE 上进行了微调。GLUE 是一组用于自然语言理解的分类任务（第 2.1.5 节）。如果忽略用于检验语言可接受性的 CoLA 任务，那么与 RoBERTa 原始结果（平均为 88.7%）相比，词序打乱后模型的平均准确率仅下降 3.4%。作者由此得出结论：类 BERT 模型几乎完全依靠利用高阶词共现统计，在下游任务中取得高性能。

常识知识的另一个侧面是时间。当预训练语言模型应用于新文档时，往往不了解新命名实体和概念的意义 [30]。模型通常无法推断文档的时间和区域，也可能无法正确整合来自不同时期或地理区域的文档中的事实。一项用于评估预训练语言模型在对话中时间推理能力的基准测试表明，BERT 和 T5 在这项任务上存在明显不足 [47]。总而言之，可以预期，能够在语言生成过程中进行检索的新型 Retro（第 6.2.3 节）或 WebGPT（第 6.2.3 节）模型，将大幅缓解本节讨论的问题。

为了能够以标准化的方式检查多种不同类型的知识，人们开发了诸如 BIG-bench 这样的大型基准测试（第 4.1.4 节）。它涵盖了常识、情商、伦理学、事实核查、一般知识、人文学科、数学、医学、阅读理解、科学和社会科学方面的基准测试。图 4.1 展示了参数量为 280B 的 Gopher 模型在这些基准测试组上的性能。在大多数组别中，准确率都超过了 50%。参数量为 540B 的 PaLM 模型能够进一步提升这些表现

数据。在约 2/3 的这些任务中，PaLM 使用 5-shot提示取得了优于人类平均水平的表现 [9, p. 17]。这表明 PaLM 的常识知识远胜于早期模型。尽管如此，PaLM 仅在少数情况下超越了人类专家的表现，这说明其仍有很大的提升空间。

一个有趣的想法是使用大型预训练多语言语言模型作为多语言知识库 [25]。作者评估了这一形式的BERT模型（第3.3.1节），这是一种标准的BERT模型，使用来自104种语言的非平行维基百科文本进行掩码语言建模损失预训练。作者发现，对于许多语言，都可以检索到正确的实体。然而，英语与日语、泰语等语言之间存在明显的性能差距。这表明，mBERT并不是以与语言无关的方式存储实体知识。如果能使用PaLM等最新语言模型重复这些实验，将会很有启发性。

4.2.3 逻辑一致性

如果一组陈述无法同时为真，则称其在逻辑上不一致。例如，考虑以下陈述：“John是Tom的父亲。Tom是John的女儿。”有时，BERT无法进行推理，即无法在逻辑上连接不同的知识片段。例如，它能够复现人物可以走进房屋以及房屋很大的关系，但无法推断出房屋比人物更大 [15, 52]。然而，对于参数更多的模型，语义知识问题往往较少。

Richardson 等人 [52] 提出了九种不同类型的简单句子对，其中包含否定、量词、比较级等。这些句子表达逻辑蕴涵、矛盾或中立。此外，他们还使用上位关系链，例如 贵宾犬 ≤ 狗 ≤ 哺乳动物 ≤ 动物，并利用这些关系生成表达相应逻辑性质的新句子。结果表明，在“逻辑任务”SNLI 和 MNLI 上进行微调的 BERT，仅在 47.3% 的情况下预测出正确陈述。

Ribeiro 等人 [51] 提出通过第 4.3.1 节所述的 *CheckList* 流程 生成大量简单示例，以测试关系。例如，该流程会测试：否定一个正面情感表达是否会导致负面情感评分。在针对商业模型和开源模型的测试中，他们观察到，超过一半的测试失败率高于 50%。

即使是更大的模型 GPT-3 也并非完美，例如，它会错误回答一些常识物理问题，比如“如果我把奶酪放进冰箱，它会融化吗？”[7]。此外，它在逻辑推理方面也有困难，例如，判断一个句子是否蕴含另一个句子。如果某个问题没有出现在它的训练材料中，GPT-3 会编出最有可能的答案，有时这个答案是错误的，例如“问题：太阳有多少只眼睛？”“回答：太阳有一只眼睛。”或“问题：*1600* 年谁是美国总统？”“回答：*1600* 年伊丽莎白一世女王是美国总统。”[29]。再来看下面这个输入“你给自己倒了一杯

蔓越莓汁，但随后你心不在焉地往里面倒入了大约一茶匙葡萄汁。看起来没问题。你试着闻一闻，但你得了重感冒，所以什么也闻不到。你非常口渴。所以你.....。GPT-3生成的续写是“把它喝了。你现在死了。”。GPT-3错误地认为“葡萄汁”是毒药，喝下去会致死 [36]。

提高逻辑一致性

如果使用适当生成的文本表达式进行训练，预训练语言模型可以提升逻辑推理能力。Richardson 等人通过使用包含否定、上位关系等内容的创建句子对 BERT模型进行微调，并使用其他生成句子进行测试， [52] 达到了 98% 的准确率。这种方法类似于第 3.6.6 节提出的数据生成策略。

类似地，Clark 等人 [10] 生成了（上下文、陈述、回答）形式的数据集，其中上下文包含不同的逻辑事实和规则，陈述是需要证明的逻辑问题，而回答为 T 或 F。随后，事实、规则和问题陈述均以（合成）英语表达。这些问题要求同时考虑多个不同的陈述才能得出结论，深度从 0（简单查找）到 5 不等。在使用这些数据进行微调期间，RoBERTa 被训练为将问题回答为真或假。在测试数据上，RoBERTa 能够以 99% 的准确率回答问题。如果对事实和规则进行释义，准确率会下降到 66%。然而，通过使用释义后的规则进行训练，该模型的准确率再次达到 90% 以上。Clark 等人 [10] 认为，通过这种方法，可以将 Transformer 视为一种“软定理证明器”，能够处理语言中的陈述。

可以将语言模型隐式的预训练知识与自然语言中的显式陈述结合起来。Talmor 等人 [64] 表明，使用此类数据集训练的模型能够进行涉及隐式世界知识和分类学知识（例如 WordNet 层级结构）的推理。此外，模型还会利用示例提供的推理模式来解决逻辑问题。

此前已有多种将逻辑推理与神经网络相结合的方法。如果神经网络能够为逻辑事实提供概率，我们就可以使用概率推理系统来施加额外约束。例如，*DeepProblog* [35]通过神经谓词融入深度学习，即由神经网络确定概率的陈述。另一种方法是概率软逻辑 （*PSL*） [28],，它支持在关系领域中进行一阶概率推理。然而，预训练语言模型无法直接为事实提供概率。已有方法尝试将自然语言句子转换为逻辑陈述，再应用逻辑推理。然而，这种“语义解析” [24] 并不十分成功。

许多研究人员已经开发了用于神经定理证明的方法。这项工作结合了符号方法和神经方法，用于推理从语言中得出的结果。例如，Minervini 等人 [39], 通过使用端到端可微模型，将逻辑谓词和文本联合嵌入共享空间，

或者 Weber 等人 [70]将 Prolog 证明器与语言模型相结合，以对自然语言应用基于规则的推理。**DeepCTRL** 方法 [57] 将规则与深度学习相结合。该方法具有一个规则编码器，可以在推理时控制规则的强度。它可应用于医疗保健、物理模型或会计等领域，在这些领域中，遵守规则至关重要。

提高逻辑一致性的一种简单而有效的方法，是增加用于构建基础模型的模型参数数量。BIG-bench 基准中很大一部分任务都用于检验逻辑一致性，例如涉及类比推理和逻辑推理的基准测试分组。*Gopher*（第 3.1.2 节）是一个拥有 280B 个参数的语言模型。它被应用于约 150 个基准测试，其中包括 19 项逻辑推理任务。在除 4 项以外的所有基准测试中，它都能提升最先进水平，这表明规模更大的预训练语言模型具有更强的推理能力。不过，其平均准确率仅约为 50%。目前尚未评估最近推出的、可检索额外文本文件的语言模型 *Retro*（第 6.2.3 节）能否改善这些结果。

PaLM（第 3.1.2 节）是一个规模更大的语言模型，拥有 540B 个参数。在 SuperGLUE 的逻辑任务 CB、COPA 和 RTE 上，与 BERT 相比，它能大幅提高得分；例如，在 COPA 上，得分从 70.6 提高到 99.2（表 4.2）。它已在数百个基准测试上接受评估，其中包括用于评估 Gopher 的那些测试。它采用了一种新的提示技术来提出逻辑问题：向系统提供示例，并附上思维链，将一项推理任务划分为更小的问题（第 3.6.4 节）。图 2.21 展示了两个示例。请注意，k-shot 推理只需为训练样本提供一个由 k 思维链提示组成的序列。随后，模型会为每个测试样本生成一条思维链。这可用于错误分析，并解释模型行为。

通过这项技术，PaLM 能够达到或超过被要求解决该任务的普通人类的性能水平。例如，可以考虑*StrategyQA*基准 [16]，，其中包含诸如“亚里士多德使用过笔记本电脑吗？”这样的问题。对于这个问题，模型必须收集有关亚里士多德寿命以及第一台笔记本电脑发明年份的事实，才能得出“否”这一回答。不使用思维链提示时，PaLM 达到了 69%；而使用思维链提示后，先前的最先进水平可从 70% 提高到 73.9%。作为比较，普通人类的准确率为 62.9%，而人类专家的准确率为 90%。

还有其他方法可以利用此类中间输出来改进学习。Wang 等人 [69]利用推理路径的多样性，对多条思维链进行采样，然后返回其中最一致的最终答案。由于为大量示例获取思维链的成本很高，Zelikman 等人 [71]在少样本设置下通过引导模型，为大型数据集生成解释，并仅保留能够得到正确答案的思维链。

4.2.4 总结

预训练语言模型拥有海量参数，能够表示极其丰富的句法知识和事实知识。这些知识可以通过探测分类器、预测掩码词、为输入生成答案或解决基准任务来加以激发。

就句法知识而言，GPT-3 等基础模型能够生成几乎没有错误的文本，并且“掌握”了大量句法规则。一个问题是如何恰当地反映否定的影响。

即使是 BERT 这样较小的模型，也能够生成大量常识知识。在这里，替换姓名或使用释义的影响存在问题。GPT-3 等规模更大的语言模型具有更强的鲁棒性，而近期提出的检索增强语言模型（WebGPT、Retro）能够为当前任务纳入相关的外部文档。这些信息可以大幅减少错误。然而，目前还没有全面的评估。一个问题是如何正确确定信息的时空定位。在这方面，BERT 和 T5 等较小模型存在很大缺陷。与此同时，在 BIG-bench 的常识知识测试中，基础模型有三分之二的得分超过了人工平均得分。由于 PaLM 等模型覆盖多种语言，它们甚至可以被用作多语言知识库。

推理的逻辑一致性是一个问题，而预训练语言模型（PLM）经常给出看似合理却错误的答案。这些模型只能针对训练文本中提及的关系进行逻辑推理，而且往往无法进行抽象，也无法将观测到的关系推广到同一类型的其他物体或实体。通过生成包含假设及其所产生的有效逻辑后果的额外训练文本，可以提高逻辑一致性。将逻辑推理系统直接纳入基础模型并未取得很大成功。参数量为 5400 亿的 PaLM语言模型通过使用思维链提示，显著提高了逻辑推理的准确率。在少样本提示中，逻辑推导会被拆解为更小的逻辑子步骤。目前尚不清楚检索增强语言模型在多大程度上能够弥补逻辑推理方面仍然存在的不足。

4.3 基准测试的可迁移性与可复现性

在本节中，我们探讨基准测试是否真正评估了其预期要测试的性质。我们还讨论其可复现程度。

4.3.1 基准测试结果的可迁移性

在一些基准测试中，基础模型的性能超过了人工标注员。这表明模型已经学到了有价值的语言知识。然而，Ribeiro 等人 [51] 认为，这可能会造成误导，因为测试集往往未涵盖恰当的内容。尽管留出测试数据上的性能是一项有用的衡量指标，但这些数据集往往不够全面。因此，我们可能会高估模型在实际应用中的可用性。

基准测试可能无法检验所有方面

在用于检测释义的 GLUE 基准中的 MRPC 任务上，RoBERTa、$\text{BERT}_{\text{LARGE}}$ 和人类的 F1 分数分别为 90.9% [34], 89.3% [42] 和 86.3%。因此，两个模型的表现都优于人类。为了检验模型是否遵循基本逻辑关系，Ribeiro 等人 [51] 提出使用**CheckList 流程**生成大量简单示例。这种方法类似于通过在单元测试中系统地生成大量不同的输入来测试软件。

例如，可以使用以下方案检查否定对情感分类任务的影响：“我 <否定> <正面_动词> 这个 <东西>”。它会生成类似“我不喜欢这种食物”或“我不喜欢帆船运动”这样的句子。作者设计了最小功能测试，用于检查模型是否确实检测到了结果的原因，或是否使用了某种缺乏依据的关联。此外，他们还利用不变性测试来查明中性扰动或释义是否会改变结果。最后，他们创建了方向性预期测试，其中的修改已知会以预期的方式改变结果。

对于 MPRC，结果表明，在这 23 个测试模板中，RoBERTa 和 BERT 分别有 11 个和 14 个模板的失败率超过 50%。因此，应谨慎看待这两个模型的“超人类”性能。

作者还测试了五种当前的预训练语言模型：$\text{BERT}_{\text{BASE}}$、$\text{RoBERTa}_{\text{BASE}}$、微软的文本分析、Google Cloud 自然语言和亚马逊的 Comprehend。他们报告了情感分类的 17 项测试结果，其中大多数问题出现在否定表达上。例如，下面的示例 “我原以为这架飞机会很糟糕，但事实并非如此。” 被大多数模型错误分类。对于包含 23 个测试模板的释义检测，难度也非常高。在这里，RoBERTa 有 11 个测试模板、BERT 有 14 个测试模板的失败率超过了 50%。在使用逻辑模板生成测试用例的阅读理解任务中，也观察到了类似的失败率。这些结果表明，基准测试原始测试集中的示例过于简单。

为了提高预训练语言模型的鲁棒性，可以生成对抗样本[8, 65]。作者讨论了使用对抗样本扩充训练数据的方法，以及生成鲁棒性证书的方法。他们还

研究了避免伪相关的方法，即在特定数据集上效果良好但通常并不成立的预测模式。

Talman等人 [63] 考察了基准测试的结果能否迁移到相似的数据集上。他们在不同的自然语言推理（自然语言推理 （*NLI*）基准测试上训练了六个预训练语言模型，这些基准测试包含人工标注为蕴含、矛盾和中性的句子对。虽然六个模型在测试集与训练集相匹配时表现良好，但使用来自其他基准测试的测试集时，准确率会显著降低。例如，$\text{BERT}_{\text{BASE}}$在 SNLI 上的测试准确率为 90.4%，而在其他基准测试的测试集上，准确率平均下降了 21.2%。造成这一降幅的原因是任务定义略有不同，文档所属领域也存在细微差异。显然，不能指望预训练语言模型的性能能够直接迁移到新数据上。

基于相关性的逻辑推理

*Winograd*模式挑战（WNLI）由 Levesque等人 [32] 提出，是 GLUE基准集合的一部分。该测试包含一对仅相差一个单词的句子，每个句子后都跟着一个问题 [41],例如

- 跑车超过了邮政卡车，因为它开得更快。问题：跑车和邮政卡车，哪辆开得更快?
- 跑车超过了邮政卡车，因为它开得更慢。问题：跑车和邮政卡车，哪辆开得更慢?

在这组句子中，一个词的差异改变了代词所指的事物或人物。要正确回答这些问题，似乎需要常识推理和世界知识。此外，作者将这些问题设计成“防谷歌搜索”的：系统不应能够通过网络搜索（或类似手段）来回答问题。GPT-3 通过少样本提示，在未进行微调的情况下达到了 88.6% 的值 [7]；DeBERTa 则在微调后达到了 95.6% 的准确率 [19]。这一准确率大致相当于人类表现。

正如 Mitchell [41] 所指出的，这并不一定意味着神经网络语言模型已经具备了类人理解。对于一些问题对，模型似乎可以通过某种相关性而非真正的世界知识来回答问题。如果模型是在大型语料库上预训练的，它就会学到上述例子中“跑车”与“快” 之间，以及“邮政卡车”与“慢” 之间的高度相关性。因此，仅凭这些相关性，而不诉诸任何理解，它就能正确判断“它” 的共指对象。结果发现，许多 Winograd模式挑战中的问题都遵循这一模式。类似的论点认为 [6, 37] ，模型可能会通过启发式方法接受某个假设，即假定前提蕴含所有词语都出现在前提中的任何假设。这意味着模型无需“理解”所讨论的情境，也能给出正确答案。

为了弥补 Winograd 模式挑战的不足，研究人员通过众包创建了规模大得多的 *Wino-grande* 基准测试 [55]。研究人员剔除了

那些可以通过利用直觉和相关性来回答的句子。他们使用 RoBERTa 创建的嵌入（第 3.1.1 节）来判断这些嵌入是否强烈指向正确的响应选项。如果是，他们就剔除这组问题对，最终得到 4.4 万个句子。下面是一个不存在相关性问题的问题对示例：

- 奖杯放不进棕色的手提箱，因为它太大了。（它：奖杯）
- 奖杯放不进棕色的手提箱，因为它太小了。（它：手提箱）

人类的准确率达到 94%，而表现最好的预训练语言模型——例如标准模型 RoBERTa——准确率仅为 79.1%。最近，*T5-XXL* 的准确率达到约 91% [43]，而拥有 269B 个参数（第 3.5.2 节）的混合专家模型 *ST-MoE-32B* [73] 取得了 96.1% 的准确率，大幅减少了错误。看来，在大多数情况下，后者能够进行“推理”，而不只是简单地关联陈述。

4.3.2 自然语言处理领域已发表结果的可复现性

许多自然语言处理领域的论文声称，其模型在某项基准测试中达到了最先进水平。例如，用于语言理解的 GLUE 基准 [67]和用于阅读理解的 SQuAD 数据集 [50]。这种方法存在两个主要问题。首先，很难评估结果是否可复现且具有统计显著性。正如 Crane [11] 所展示的，通常有许多未报告的条件会影响结果的可复现性。例如，网络参数的随机初始化。由此产生的方差往往大于所报告的最先进水平得分提升。然而，通常不会报告这些因素所导致的方差。其他影响因素包括底层编程框架和库，它们会随着时间而变化。超参数、预处理细节和模型配置通常也未予说明。

为记录模型架构、模型的训练与评估过程，Mitchell 等人 [40]提出在**模型卡**中描述相关事实和超参数。对模型及其用途进行简短的高层描述后，模型卡应包含九个不同的部分 [40]：

1. 模型的基本信息，2. 预期用途和范围限制，3. 模型在各种相关因素下的性能，4. 性能指标，5. 评估数据，6. 训练数据，7. 根据所选指标得出的评估结果。8. 伦理考量、风险和危害。9. 注意事项和建议。

HuggingFace 提供了更多详细信息 [22]。即使模型仍然可以在没有模型卡的情况下发布，对模型进行明确的文档记录也只会让未来的用户受益。因此，应尽可能提供模型卡。对于大多数近期模型，即使模型不是开源的，也会提供模型卡。

Belz 等人 [4] 对自然语言处理中的可复现性 进行了综述。他们指出，性能结果往往取决于模型参数和设置中一些看似微小的差异，例如罕见词的最小计数或文本规范化。作者在一项重复实验研究中指出，报告的 513 个得分中，只有 14% 完全相同。令人遗憾的是，有 59% 的得分低于已发表的数值。因此，论文中发表的实验结果应谨慎看待。

另一个问题是，性能提升究竟由什么导致。如上文所述，增加预训练语言模型的参数数量和计算开销通常会带来更好的结果（第 3.5.1 节）。因此，模型的架构变更究竟提升了性能，还是仅仅因为参数数量增加或训练集规模扩大[53]，往往并不明确。

显然，使用更大的模型和更多的计算开销，就能在排行榜上取得第一名。不过，根据 Rogers [53] 的说法，这“算不上研究新闻”。此外，这些结果往往无法复现：谁负担得起花 460 万美元重新训练 GPT-3？因此，开发更小但更具创新性的模型回报较低，因为很难击败更大的模型。只有当新模型的作者能够证明，在参数数量和计算预算相同的情况下，其架构优于此前的最先进模型时，才能声称自己做出了有价值的贡献。Rogers [53] 提议为排行榜提供标准训练语料库，并将计算开销限制在强基线模型的水平。另一种做法是报告训练数据的规模和计算开销，并将其纳入最终得分。

可用实现

- HuggingFace 上提供了 RoBERTa 和 ELECTRA 的模型代码及训练模型：https://huggingface.co/transformers/。
- DeBERTa 的代码可在 https://github.com/microsoft/DeBERTa 和 Hugging Face 获取。
- CheckList 的代码位于 https://github.com/marcotcr/checklist。

4.3.3 总结

基准测试结果的可迁移性并不总能延伸到实际应用。即使预训练语言模型在测试集上的逻辑推理能力优于人类，也可能无法

正确地对生成的逻辑推理链进行分类。这表明测试集未涵盖所有可能示例的完整范围。在相关基准测试中的性能通常会更低，因为领域或任务的定义可能存在偏差。

有时，逻辑结论并非通过逻辑演绎得出，而是通过对先行词和后件进行简单的相关性分析得出。GLUE 基准中的 Winograd 任务就可以说明这一点。为避免这种类型的“推理”，研究人员开发了一种名为 Winogrande 的新变体任务，其中相关性与推理任务无关。与此同时，一个拥有 269B 参数的基础模型也能够比人类更好地解决这一任务。

一项关于自然语言处理中结果可复现性的综述表明，已发表的性能往往取决于一些未报告的因素，例如随机数初始化。这些因素造成的变异通常大于所报告的改进。因此，有必要报告这些因素导致的方差。此外，还应在模型卡中记录模型架构、训练和评估的详细信息。在大约 500 次重复实验中，最终得分中约有 60% 低于报告值，这一比例令人不安。请注意，由于参数更多、训练数据更多或计算开销更高而带来的改进，并不能说明模型架构更好。

参考文献

[1] S. Aarohi 和 R. Abhinav。《*BIG-bench*》。Google，2022年6月20日。网址：https://github.com/google/BIG-bench/blob/936c4a5876646966344349b28ae187c556938ec4/docs/paper/BIG-bench.pdf（访问于2022年6月20日）。[2] M. Aßenmacher 和 C. Heumann。《预训练语言模型的可比性》。2020年。arXiv：2001.00781。[3] S. Balasubramanian、N. Jain、G. Jindal、A. Awasthi 和 S. Sarawagi。《姓名意味着什么？BERT 命名实体表示对任何其他姓名都同样出色吗？》2020年。arXiv：2007.06897。[4] A. Belz、S. Agarwal、A. Shimorina 和 E. Reiter。《自然语言处理领域可复现性研究的系统综述》。2021年3月21日。arXiv：2103.07929 [cs]。[5] S. R. Bowman 和 G. E. Dahl。《修复自然语言理解基准测试需要什么？》2021年。arXiv：2104.02145。[6] R. Branco、A. Branco、J. António Rodrigues 和 J. R. Silva。《捷径化常识：深度学习常识推理中的数据虚假性》。载于：《*2021*年自然语言处理经验方法会议论文集》EMNLP 2021。线上及多米尼加共和国蓬塔卡纳：计算语言学协会，2021年11月，第1504–1521页。doi：10.18653/v1/2021.emnlp-main.113。[7] T. B. Brown 等。《语言模型是少样本学习器》。2020年。arXiv：2005.14165。[8] K.-W. Chang、H. He、R. Jia 和 S. Singh。《自然语言处理中的鲁棒性与对抗样本》。载于：《*2021*年自然语言处理经验方法会议教程摘要论文集》蓬塔卡纳，多米尼加共和国及线上：计算语言学协会，2021年11月，第22–26页。网址：https://aclanthology.org/2021.emnlp-tutorials.5（访问于2021年11月24日）。[9] A. Chowdhery 等。《PaLM：借助 Pathways 扩展语言建模》。2022年4月5日。arXiv：2204.02311 [cs]。

[10] P. Clark、O. Tafjord 和 K. Richardson。《Transformer 作为语言上的软推理器》。2020年。arXiv：2002.05867。[11] M. Crane。《问答研究中的可疑答案：已发表结果的可复现性与变异性》。载于：《计算语言学协会会刊》6（2018），第241–252页。[12] Y. Elazar、N. Kassner、S. Ravfogel、A. Ravichander、E. Hovy、H. Schütze 和 Y. Goldberg。《衡量并改进预训练语言模型的一致性》。2021年5月29日。arXiv：2102.01017。[13] Y. Elazar、S. Ravfogel、A. Jacovi 和 Y. Goldberg。《失忆探测：利用失忆反事实进行行为解释》。载于：《计算语言学协会会刊》9（2021），第160–175页。[14] A. Ettinger。《BERT 不是什么：语言模型心理语言学诊断新套件带来的启示》。载于：《计算语言学协会会刊》8（2020），第34–48页。[15] M. Forbes、A. Holtzman 和 Y. Choi。《神经语言表示是否学习了物理常识？》2019年。arXiv：1908.02899。[16] M. Geva、D. Khashabi、E. Segal、T. Khot、D. Roth 和 J. Berant。《亚里士多德使用过笔记本电脑吗？带有隐式推理策略的问答基准测试》。载于：《计算语言学协会会刊》9（2021），第346–361页。[17] Y. Goldberg。《评估 BERT 的句法能力》。2019年。arXiv：1901.05287。[18] P. He、J. Gao 和 W. Chen。《DeBERTaV3：利用具有梯度解耦嵌入共享的 ELECTRA 风格预训练改进 DeBERTa》。2021年。arXiv：2111.09543。[19] P. He、X. Liu、J. Gao 和 W. Chen。《DeBERTa：通过解码增强的 BERT 与解耦注意力机制》。2021年1月11日。arXiv：2006.03654。[20] D. Hendrycks、C. Burns、S. Basart、A. Zou、M. Mazeika、D. Song 和 J. Steinhardt。《衡量大规模多任务语言理解能力》。2020年。arXiv：2009.03300。[21] J. Hewitt 和 C. D. Manning。《用于在单词表示中发现句法的结构探针》。载于：《2019年北美计算语言学协会人类语言技术分会会议论文集》第1卷：长篇与短篇论文。2019年，第4129–4138页。[22] huggingface。《构建模型卡——Hugging Face 课程》。2022年。网址：https://huggingface. co/course/chapter4/4（访问于2022年8月7日）。[23] G. Jawahar、B. Sagot 和 D. Seddah。《BERT 学到了哪些关于语言结构的知识？》载于：2019年。[24] A. Kamath 和 R. Das。《语义解析综述》。2018年。arXiv：1812.00978。[25] N. Kassner、P. Dufter 和 H. Schütze。《多语言 LAMA：研究多语言预训练语言模型中的知识》。2021年。arXiv：2102.00894。[26] N. Kassner 和 H. Schütze。《面向预训练语言模型的否定探测与错误启动探测：鸟类会说话，但不会飞》。2019年。arXiv：1911.03343。[27] T. Kim、J. Choi、D. Edmiston 和 S.-g. Lee。《预训练语言模型了解短语吗？用于语法归纳的简单但强大的基线》。2020年。arXiv：2002.00737。[28] B. Kirsch、S. Giesselbach、T. Schmude、M. Völkening、F. Rostalski 和 S. Rüping。《利用概率软逻辑改进法律领域的信息抽取》。载于：（2020年）。[29] K. Lacker。《给 GPT-3 做图灵测试》。2020年7月6日。网址：https://lacker.io/ai/2020/07/06/giving- gpt-3-a-turing-test.html（访问于2020年12月3日）。[30] A. Lazaridou 等。《注意差距：评估神经语言模型中的时间泛化》。载于：《神经信息处理系统进展》34（2021）。[31] B. Lemoine。《LaMDA 具有感知能力吗？——一次访谈》。Medium。2022年6月11日。网址：https: // cajundiscordian.medium.com/is-lamda-sentient-an-interview-ea64d916d917（访问于2022年6月24日）。[32] H. Levesque、E. Davis 和 L. Morgenstern。《Winograd模式挑战》。载于：《第十三届知识表示与推理原理国际会议论文集》。2012年。[33] N. F. Liu、M. Gardner、Y. Belinkov、M. E. Peters 和 N. A. Smith。《语言学知识与上下文表示的可迁移性》。2019年。arXiv：1903.08855。

[34] Y. Liu 等人。“RoBERTa：一种鲁棒优化的 BERT 预训练方法”。2019年。arXiv：1907.11692。[35] R. Manhaeve、S. Dumancic、A. Kimmig、T. Demeester 和 L. De Raedt。“DeepProblog：神经概率逻辑编程”。载于：《神经信息处理系统进展》2018年，第3749–3759页。[36] G. Marcus 和 E. Davis。GPT-3：常识推理。2020年8月1日。网址：https: // cs.nyu.edu/faculty/davise/papers/GPT3CompleteTests.html（访问日期：2021年2月15日）。[37] R. T. McCoy、E. Pavlick 和 T. Linzen。“错误理由下的正确答案：诊断自然语言推理中的句法启发式”。6月 24, 2019 1902 01007 。arXiv：。cs。[38] S. Merity、C. Xiong、J. Bradbury 和 R. Socher。“指针哨兵混合模型”。2016年。arXiv：1609.07843。[39] P. Minervini、M. Bošnjak、T. Rocktäschel、S. Riedel 和 E. Grefenstette。“在大型知识库和自然语言上的可微推理”。载于：《AAAI人工智能会议论文集》第34卷。04。2020年，第5182–5190页。[40] M. Mitchell 等人。“用于模型报告的模型卡”。载于：《公平性、问责制与透明度会议论文集》。2019年1月29日，第220–229页。doi：10.1145/3287560.3287596。arXiv：1810。03993 [cs]。[41] M. Mitchell。人工智能理解究竟意味着什么？《Quanta Magazine》。2021年12月16日。网址：https://www.quantamagazine.org/what-does-it-mean-for-ai-to-understand-20211216/（访问日期：2022年1月3日）。[42] N. Nangia 和 S. R. Bowman。“人类与木偶：GLUE 基准上人类表现的保守估计”。2019年6月1日。arXiv：1905。10425 [cs]。[43] openai。提交系统——WinoGrande：大规模对抗性Winograd模式挑战排行榜。2022年1月5日。网址：https://leaderboard.allenai.org/winogrande/submissions/ public（访问日期：2022年1月5日）。[44] D. Paperno 等人。“LAMBADA 数据集：需要广泛话语上下文的词语预测”。2016年6月20日。arXiv：1606。06031 [cs]。[45] Paperswithcode。浏览人工智能领域的最先进成果。2019年。网址：https://paperswithcode.com/sota。[46] F. Petroni、T. Rocktäschel、P. Lewis、A. Bakhtin、Y. Wu、A. H. Miller 和 S. Riedel。“语言模型即知识库？”2019年。arXiv：1909.01066。[47] L. Qin、A. Gupta、S. Upadhyay、L. He、Y. Choi 和 M. Faruqui。“TIMEDIAL：对话中的时间常识推理”。载于：《第59届计算语言学协会年会暨第11届自然语言处理国际联合会议论文集》第1卷长篇论文。ACL-IJCNLP 2021。在线版：计算语言学协会，2021年8月，第7066–7076页。doi：10.18653/v1/ 2021.acl-long.549。[48] A. Radford、J. Wu、D. Amodei、D. Amodei、J. Clark、M. Brundage 和 I. Sutskever。“更好的语言模型及其影响”。载于：《OpenAI博客》（2019年）。网址：https: //openai.% 20com/blog/better-language-models。[49] J. W. Rae 等人。“扩展语言模型：训练 Gopher 的方法、分析与洞见”。载于：ArXiv 预印本 ArXiv211211446（2021年12月8日），第118页。[50] P. Rajpurkar、J. Zhang、K. Lopyrev 和 P. Liang。“SQuAD：100，000+ 个用于机器阅读理解文本的问题”。2016年。arXiv：1606.05250。[51] M. T. Ribeiro、T. Wu、C. Guestrin 和 S. Singh。“超越准确率：使用 CheckList 对自然语言处理模型进行行为测试”。2020年。arXiv：2005.04118。[52] K. Richardson、H. Hu、L. Moss 和 A. Sabharwal。“通过语义片段探测自然语言推理模型”。载于：《AAAI人工智能会议论文集》第34卷。05。2020年，第8713–8721页。[53] A. Rogers。Transformer 如何打破自然语言处理排行榜。Hacking semantics。2019年6月30日 2019 12 15 2021。网址：https://hackingsemantics.xyz/ /leaderboards/（访问日期：//）。[54] A. Rogers、O. Kovaleva 和 A. Rumshisky。“BERT学入门：我们对 BERT 工作方式的了解”。载于：《计算语言学协会会刊》8（2021年），第842–866页。[55] K. Sakaguchi、R. L. Bras、C. Bhagavatula 和 Y. Choi。“WinoGrande：大规模对抗性Winograd模式挑战”。载于：《ACM通讯》64.9（2021年），第99–106页。

[56] F. Schmidt 和 T. Hofmann。“BERT 作为教师：序列级奖励的上下文嵌入”。2020年。arXiv：2003.02738。[57] S. Seo、S. Arik、J. Yoon、X. Zhang、K. Sohn 和 T. Pfister。“使用规则表示控制神经网络”。载于：《神经信息处理系统进展》34（2021年）。[58] M. Shoeybi、M. Patwary、R. Puri、P. LeGresley、J. Casper 和 B. Catanzaro。“Megatron-LM：使用模型并行训练数十亿参数语言模型”。载于：arXiv（2019年），arXiv–1909。[59] K. Sinha、R. Jia、D. Hupkes、J. Pineau、A. Williams 和 D. Kiela。“掩码语言建模与分布假设：词序很重要——针对少量预训练的研究”。2021年4月14日。arXiv：2104.06644。[60] J. Sohl-Dickstein。BIG-bench。Google，2021年12月16日。网址：https://github.com/google/BIG-bench（访问日期：2021年12月16日）。[61] M. Sparkes。Google 希望用200项任务挑战人工智能，以取代图灵测试。《新科学家》。2022年6月14日。网址：https://www.newscientist.com/article/2323685-google-wants-to-challenge-ai-with-200-tasks-to-replace-the-turing-test/（访问日期：2022年6月26日）。[62] S. Storks、Q. Gao 和 J. Y. Chai。“自然语言理解的常识推理：基准测试、资源与方法综述”。2019年。arXiv：1904.01172。[63] A. Talman 和 S. Chatzikyriakidis。“跨自然语言推理基准测试神经网络模型的泛化能力”。2019年5月31日。arXiv：1810.09774。[64] A. Talmor、O. Tafjord、P. Clark、Y. Goldberg 和 J. Berant。“教导预训练模型系统性地推理隐性知识”。2020年。arXiv：2006.06609。[65] TrustworthyAI，导演。CVPR 2021 教程：“深度学习中的实用对抗鲁棒性：问题与解决方案”。2021年6月28日。网址：https://www.youtube.com/ watch?v=ZmkU1YO4X7U（访问日期：2022年2月26日）。[66] Wang。SuperGLUE 基准。SuperGLUE 基准。2021年。网址：https : / / super . gluebenchmark.com/（访问日期：2021年2月23日）。[67] A. Wang、A. Singh、J. Michael、F. Hill、O. Levy 和 S. R. Bowman。“GLUE：用于自然语言理解的多任务基准与分析平台”。2019年2月22日。arXiv：1804.07461。[68] A. Wang 等人。“SuperGLUE：面向通用语言理解系统的更具挑战性的基准”。载于：《神经信息处理系统进展》2019年，第3266–3280页。[69] X. Wang 等人。“自洽性提升语言模型中的思维链推理”。2022年4月6日。arXiv：2203。11171 [cs]。[70] L. Weber、P. Minervini、J. Münchmeyer、U. Leser 和 T. Rocktäschel。“NLProlog：利用弱合一进行自然语言问答推理”。2019年。arXiv：1906.06187。[71] E. Zelikman、Y. Wu 和 N. D. Goodman。“STaR：通过推理引导推理”。2022年3月27日。arXiv：2203。14465 [cs]。[72] X. Zhou、Y. Zhang、L. Cui 和 D. Huang。“评估预训练语言模型中的常识”。载于：AAAI。2020年，第9733–9740页。[73] B. Zoph 等人。“设计有效的稀疏专家模型”。2022年。arXiv： 2202.08906。

第5章
用于信息抽取的基础模型

摘要 本章探讨信息抽取方法，这些方法能够自动识别文本文件中的结构化信息，并涵盖一系列任务。文本分类任务是将文档归入一个或多个预先定义的内容类别或类别中。这包括许多子任务，例如语言识别、情感分析等。词义消歧任务为文档中的每个单词指定预先定义的意义。命名实体识别任务用于识别文档中的命名实体。实体是文本中提及的任何对象或概念，而命名实体则是以专有名称指称的实体。关系抽取任务旨在识别从文本中提取的实体之间的关系。这涵盖了许多子任务，例如共指消解、实体链接和事件抽取。最具挑战性的是从文本中联合抽取实体和关系。传统上，相对较小的预训练语言模型经过针对这些任务的微调后，性能很高；而较大的基础模型通过少样本提示也能取得高分，但通常尚未经过基准测试。

关键词： 文本分类、命名实体识别、关系抽取、情感分析、语言理解

预训练语言模型（PLM）有大量自然语言处理应用，大致可以分为三个领域

- 信息抽取 （*IE*）能够自动识别文档中的结构化信息，并分析语言特征（第 5 章）。
- 自然语言生成 （*NLG*）能够自动生成新的自然语言文本，通常用于响应某个提示（第 6 章）。
- 多模态内容分析 与生成整合了文本、语音、图像、视频等两种或更多模态的内容理解与生成（第 7 章）。

这些应用将在接下来的三章中进行介绍。

在本章中，我们聚焦于使用预训练语言模型进行**信息抽取**。信息抽取包括以下任务：

- 文本分类是将文档归入一个或多个预定义的内容类别（第5.1节）。请注意，许多子任务都可以表述为分类问题，例如语言识别、情感分析等（表5.1）。
- 词义消歧（*WSD*）将预定义的意义与文档中的每个单词关联起来。这对于理解同形异义词尤为重要，即那些会因上下文不同而具有多种意义的词语（第 5.2 节）。
- 命名实体识别（*NER*）用于识别文档中的命名实体。实体是文本中提及的任何对象或概念。命名实体是由专有名称指代的实体。命名实体识别还会为每个实体标注类型，例如人、位置、组织等（第 5.3 节）。
- 关系抽取旨在识别从文本中抽取的实体之间的关系（第 5.4 节）。这涵盖许多子任务，例如共指消解、实体链接和事件抽取（表 5.3）。

由于方法种类繁多，我们聚焦于在撰写本文时性能较高的代表性模型。传统上，人们会对相对较小的预训练语言模型进行微调，以完成这些任务并取得较高性能；而较大的基础模型则通过少样本提示取得较高得分，但通常尚未经过基准测试。

我们概述这些方法的内在逻辑和主要特征，同时考虑必要的资源，例如计算和内存需求。对于标准模型，我们提供了前几章中相关描述的链接。在“可用实现”标题下，您可以找到某项任务的可用代码和预训练模型链接。有关代码的优质通用来源是网站 [30, 35, 74, 79]。

5.1 文本分类

自动文本分类是自然语言处理中一项常见任务，即为短文本或文档分配一个类别（也称为类别或标签）。类别集合是预先定义的，可以只包含两个类别（二元分类），也可以包含更多类别（多类分类）。每段文本都必须分配一个类别，这意味着这些类别彼此互斥。典型任务包括电子邮件中的垃圾邮件检测、情感分析、新闻文章分类、仇恨言论检测、对话行为分类等。表 5.1 列出了一些示例。Kowsari 等人 [44], 、Li 等人 [49] 和 Minaee 等人 [64] 对文本分类进行了综述。

一篇文档通常会同时涉及多个主题，例如一篇关于足球场馆建设成本的新闻文章。在这种情况下，需要为文档分配多个类别，例如本例中的“足球”和“金融”。这种分类称为多标签分类。极端多标签分类是一种变体，其标签集非常庞大，包含数千个标签。

表 5.1 基于文本分类的语言分析任务示例。

Task	描述	示例
语言识别	确定文本的语言，第 1.2 节。	莎士比亚生活在 *400* 年前 → 英语
文档分类	为内容类别（类别）分配一个类别，例如经济，分配给文档或文本，第5.1节	道琼斯上涨了*50*个得分 → 经济
情感分析	根据文本进行分类其中表达的情感（例如正面、负面、中性），第5.1节	我今天感觉特别糟糕。 → 负面
仇恨言论检测	识别文本是否包含仇恨言论，第5.1.1节	移民侵扰我们的国家 → 仇恨言论
虚假新闻检测	检测包含假新闻的文本，第6.5.5节	麻疹疫苗接种会导致脑膜炎。 → 假新闻
逻辑关系	判断第二段文本是否包含一个逻辑后果、一个矛盾，或一个中性陈述相对于第一段文本，第 2.1.5 节	*John* 有一套公寓。 $\leftrightarrow_{contradiction}$ *John* 是一名无家可归者。
文本蕴涵	第一段文本是否蕴含第二段文本为真？第 2.1.5 节	锻炼有益于健康。 $\rightarrow_{entails}$ 身体活动有益后果。
释义检测	判断两段文本是否语义等价，第 2.1.5 节	弗雷德累了。/弗雷德想睡觉。 → 等价
对话行为分类	确定……中的话语类型对话（问题、陈述、请求执行动作等）	狗在哪里？ → 问题

有许多常用的基准测试，用于评估涵盖两个或更多类别的文档分类方法的性能。通常，这些基准测试包含数千个训练样例和测试样例。表 5.2 描述了一些常用文本分类基准测试的性质。文档通常根据用户的主观观点进行分类。例如，电影或餐馆的评论可以分为正面、负面或中性。此时，这种分类对应于情感分析任务。

20世纪90年代早期的文档分类方法采用了经典的机器学习方法 [44]。在预处理的第一步，从文档中提取人工创建的特征。第二步，使用这些特征训练分类器，以重现人工分配的类别标签（第 1.3 节）。最后，将该分类器应用于新的文档。通常使用词袋 表示来表示输入文档。常用的分类方法包括朴素贝叶斯、逻辑分类器、支持向量机以及随机森林等基于树的方法。然而，所有这些方法都受到词袋表示的缺陷所限（第 1.3 节），因为这种表示忽略了文档中词语的顺序。

表 5.2 热门文本分类基准测试。

Task	描述	类别
IMDB [56]	来自电影评分网站 IMDB 的评论。25k 训练数据、25k 条测试评论和 50k 条未标注评论	两个类别：正面和负面
Yelp [131]	关于商店和餐馆的 Yelp 评论。560k 训练数据和 38k 条文本评论。	二分类：正面 / 负面 多分类：五星级类别
DBpedia [7]	DBpedia 中 14 个互不重叠的类别本体。每个类别由 4 万个训练样本和 5,000 个测试样本组成，	14 个不同类别：公司、艺术家、运动员，动物、专辑、电影等。
ArXiv [32]	来自 arXiv 的 3.3 万篇科学文章，包含平均长度为 6,300 的文档，长度 > 5，000	11 个类别：人工智能、计算机视觉、群论等。
SemEval-20 任务 12 [128]	可供五种语言使用的 1.4 万条 Twitter 推文：英语、阿拉伯语、丹麦语、希腊语、土耳其语	两个类别：冒犯性或非冒犯性。
EURLex-4K [53]	包含法律文件的基准测试，包含 45,000 个训练样本，平均长度为 727 个词，平均为五个每个示例的正确类别	4,271 个非互斥类别
Amazon670k 数据集 [60]	亚马逊产品的描述。490k 训练数据以及 153k 个测试样本。每份文档约有 5.5 个类别。	679k 个非互斥类别：亚马逊目录中的产品，约有 4 个样本/类别

接下来的章节中，我们将探讨适用于互斥类别以及“重叠”类别的当前分类模型。事实证明，目前大多数最佳方法都基于预训练语言模型。

5.1.1 采用互斥类别的多类分类

BERT 的一个重要应用是针对分类任务进行微调（第 2.1.2 节）。在此过程中，通过有监督微调，将预训练BERT适配到该任务：使用最高层中“[*CLS*]”词元的上下文嵌入作为逻辑分类器的输入。该分类器在自然语言理解任务（第 2.1.5 节）中取得了极佳的效果。

XLNet [120] 通过重构置换后的词元序列（第 3.1.1 节）进行训练，因此能够捕捉大量关于语言的知识。在 IMDB 二分类任务中，其准确率达到 96.2%。这一性能被**ERNIE-Doc** [26] 以 97.1% 的准确率超越。ERNIE-Doc 是一种采用增强循环机制的 Transformer，能够在同一层中考虑文本的许多前置片段。该模型旨在缓解 Longformer 等其他基于 Transformer 的模型在长上下文中的问题；这些模型无法为每个片段提供整个文档的上下文信息。目前，最先进水平是

由一种更简单的模型保持，该模型修改了广为人知的段落向量 [47]以及朴素贝叶斯加权的 *n*-gram。其准确率达到 97.4%。

目前，在包含 10 个类别的 IMDB数据集上表现最佳的模型是 **ALBERT-SEN**[23]。作者提出了一种方法，用于评估句子对整个文档的重要性，其动机是：不同句子可能包含不同的极性，但整体极性取决于少数几个重要句子。该模型使用 ALBERT（第 3.1.1 节），通过[*SEP*] 和 [*CLS*] 词元表示来编码句子。随后，将这些表示与按类别加权的表示拼接起来。然后，模型利用文档编码器计算每个句子的重要性权重，并生成句子的加权表示，作为文档表示。最后，模型利用文档表示和类别表示计算情感得分；类别表示也用于句子编码器。该模型的准确率达到 54.8%。值得注意的是，在这个包含 10 个类别的分类任务中，必须考虑语言表达中的细微差别。

在 Yelp 基准测试中，**XLNet** 在二分类版本上表现最佳，准确率为 98.4%；在包含 5 个类别的细粒度版本上，其准确率为 73.0%，排名第二。该任务的领先模型是**HAHNN** [1]，准确率为 73.3%。HAHNN 将卷积层、门控循环单元和注意力机制结合起来。它以非上下文化的 FastText [16] 嵌入作为词表示，并通过堆叠卷积层来获取上下文信息。接下来，词编码器应用循环 GRU 单元来获得词表示，并通过注意力机制为输入词生成权重。随后，将词的表示加权平均，形成句子表示。模型还使用另一个 GRU 层生成句子表示，再通过注意力机制将其组合，生成文档级表示。该表示作为全连接层的输入，并通过 Softmax 激活函数完成分类。

BigBird [127] 对于长文档的分类任务尤其有价值，因为它可以处理长度为 4,096 的输入序列（第 3.2.1 节）。遵循 BERT 的做法，第一个 [*CLS*] 的输出嵌入将作为分类器的输入。对于包含较短文档的 IMDB 数据，与更简单的模型相比，BigBird 并未带来性能提升。在平均长度为 6300、包含 11 个类别的 *ArXiv*基准 [32] 上，BigBird 将 Sota 提高了约 5 个百分点。

随着 Web 2.0 的出现，以及用户能够创建自己的内容并与全世界分享，这类仇恨言论等有害内容日益增多。如今，机器人和机器学习模型进一步推动了这一趋势，它们能够自动生成此类内容，其规模之大几乎超出人类的处理能力。仇恨言论 通常被定义为个人或群体针对种族、肤色、国籍、性别、残障、宗教或性取向等特征所进行的敌意或贬损性交流 [36]。根据欧洲法律，仇恨言论属于应受刑罚的刑事犯罪。

仇恨言论检测可以作为文本分类任务来解决。识别此类文本很困难，因为仇恨言论、反讽、言论自由和艺术之间的界限十分模糊。Jahan 等人 [36] 和 Yin 等人 [123] 对自动

仇恨言论检测进行了系统综述。鉴于该任务的重要性，让我们更仔细地了解当前的方法。

Roy 等人 [88] 采用了一种多语言方法。他们通过对推文进行特殊分词，预处理来自 Twitter 的文本。清理后的文本、表情符号和分段的话题标签由不同的 Transformer 编码，并拼接在一起。最后，多层感知器生成分类结果。*HASOC 2019*推文数据集 [58] 的结果表明，表情符号和话题标签提供的额外信号提升了仇恨言论检测以及仇恨言论类型分类的性能。他们在英语、德语和印地语测试集上分别取得了 90.3%、81.9% 和 75.5% 的 F1 值。

Mathew 等人 [59] 认为，应当解释仇恨言论分类器的决策。他们提出了包含约 2 万篇帖子的 *HateXplain* 数据集。标注内容包括类别标签（仇恨、冒犯性或正常）、遭到诋毁的目标群体，以及导致分类结果的词语片段标注。总体而言，BERT模型在解释仇恨言论分类决策方面表现最佳。

最近的一项竞赛是 SemEval-20 任务 12 [128]，其中 14,100 条 Twitter 推文被人工标注为冒犯性或非冒犯性。使用 **RoBERTa**分类器（第 3.1.1 节），Wiedemann 等人 [110] 取得了 92.0% 的 F1值，并赢得了竞赛。在后续实验中，*ALBERT* 模型（第 3.1.1 节）的集成将该得分提高到了 92.6%。总之，借助高质量的预训练语言模型，可以解决仇恨言论的自动分类问题。

5.1.2 多标签分类

当一段文本可以同时属于多个类别时，就需要进行多标签分类。当可用类别数量非常庞大时，这有时被称为极端多标签分类。一个示例问题是为维基百科文章分配标签，维基百科拥有近 500k 个标签。在多标签分类中，通常会返回每个类别的得分或概率。这可以用来对类别进行排序。准确率等传统指标假设只有一个类别是正确的，因此无法应用。另一种方法是衡量由得分所产生的排序质量（参见第 6.1.2 节）。对于预测得分向量 $\hat{y}_i \in [0,1]$ 和真实标签向量 $y_i \in \{0,1\}$，一种常用的衡量指标是*precision atk*，它统计得分最高的 k 个类别中有多少个是正确的：

$$prec@k = \frac{1}{k} \sum_{l \in \text{rank}_k(\hat{y})} y_l \qquad DCG@k = \frac{1}{k} \sum_{l \in rank_k(\hat{y})} \frac{y_l}{\log(l+1)}, \tag{5.1}$$

其中，$\text{rank}(\hat{y}) = (i_1, \ldots\ldots, i_k)$ 是由 $\hat{y}_i$ 中最大的 k 个值按降序排列后对应的索引组成的向量 $\hat{y}_{i_1} \geq \cdots \geq \hat{y}_{i_k}$。第二个指标 $DCG@k$ 是折损累计增益，其中正确匹配 y_l 会根据其排名 l 乘以 $1/\log(l+1)$ [14] 进行加权。这反映了排名较低的正确匹配

应获得较低的权重。此外，还有一个归一化版本$nDCG@k$，其中 $DCG@k$ 除以其最大可能值。

为每个类别分别训练分类器通常能获得很高的准确率，但训练和预测速度却非常慢。在最坏情况下，必须为每个标签训练一个分类器，将该标签的所有正实例作为正样本，并将其他标签的所有实例作为负样本。为缓解这一问题，**Parabel** [83] 采用了树集成。Parabel 首先通过对属于某个标签的所有实例求均值来构造标签表示，并将所得向量归一化为单位向量。随后，在标签空间上递归应用平衡二均值聚类，直到聚类标签树中的所有叶节点都包含少于 M 个标签，例如 $M = 100$。针对树的每个内部节点和叶节点，都会训练分类器，以判断某个实例应沿树中的哪条路径前进。这样，便能根据标签表示高效地生成平衡的标签层次结构，使输入相似的标签最终聚集在一起，位于叶节点。集成中最多使用 3 棵这样的树。

最后，对每个标签，都训练一对多分类器，以估计标签上的联合概率分布的最大后验值。用于训练这些分类器的负例取自同一叶节点中的其他标签，因此采用了最相似或最容易混淆的反例。进行预测时，会在树中执行束搜索，并且只对 k 个概率最高的标签实际进行分类。Parabel 已应用于拥有 700 万个标签的问题，并能以对数时间做出预测。与最先进的极端分类器相比，Parabel 在训练和预测方面显著更快，同时精确率几乎相同。在 EURLex-4K 上，其 prec@1 值为 81.5；在 Amazon-670k 上，prec@1 值为 43.9，低于最佳方法的 45.4，但其预测耗时仅为后者的 1/1000。

AttentionXML [124] 是一种基于树的分类器，它使用上下文嵌入作为输入特征。通过在众多标签与词元之间应用注意力，AttentionXML 能够针对每个标签，以不同方式表示给定文本。AttentionXML 的架构由单词表示层、双向 LSTM层、一个注意力层（所有标签均对经 BiLSTM（第1.6节）编码的输入施加注意力），以及一个全连接层和输出层组成。

AttentionXML 首先构建一棵类似 Parabel 的深层树。然后将该树压缩为一棵浅而宽的树，从而能够处理数百万个类别，尤其是“尾部标签”，即训练集[37]中仅有少量示例的类别。该模型使用二元交叉熵损失函数。针对树的每一层，都会训练该模型，并使用前一层树所对应的模型进行初始化。AttentionXML 使用通过微调后的标签召回模型采样得到的负面标签进行标签排序训练。在预测时，模型使用该树进行束搜索，因此只考虑父节点得分最高的树分支。

在 *EURLex-4K* 基准上，AttentionXML 达到 $prec@1 = 87.1\%$ 和$prec@5 = 61.9\%$。这意味着，模型得分最高的预测中有 87.1% 正确，而得分最高的五个预测中有 61.9% 正确。请注意，应根据训练集中每篇文档的平均标签数来选择 k。在 *Amazon670k* 数据集上， [60] 面对 679k 个类别，AttentionXML 达到 $prec@1 = 47.6\%$ 和

$prec@5 = 38.9\%$。这意味着，大约 40% 的替代产品都能被正确识别。

LightXML [39] 采用 Transformer 编码器生成上下文词特征，并以动态方式为每个类别生成负例。首先，根据输入特征创建一组标签簇，使每个标签都属于一个簇。然后，采用 RoBERTa 等预训练模型（第 3.1.1 节）将一个训练样本的输入文本编码为上下文嵌入。为表示训练样本的输入文本，将最后五层中 [*CLS*] 词元的嵌入进行拼接。

特定的<label recalling 模型旨在使用 [*CLS*]嵌入作为输入来预测标签簇。此外，*label ranking model* 接收训练样本的 [*CLS*] 嵌入以及相应的标签。标签召回模型会动态生成带有其他标签的负例。生成器和判别器的损失项被合并到联合损失函数中，从而实现端到端训练。在 EURLex-4K 基准测试中，LightXML 达到 $prec@1 = 87。6\%$ 和 $prec@5 = 63。4\%$。在 Amazon670k 基准测试中，其达到 $prec@1 = 49。1\%$ 和 $prec@5 = 39。6\%$。这两个值都略高于 AttentionXML。与其他三个多标签数据集上的 7 个替代模型相比，该方法也展现出最先进水平的性能。

重叠 [51] 将标签分组成相互重叠的簇。例如，在产品分类中，标签“皮带”可以指车辆皮带（属于“车辆配件”类别），也可以指男士腰带（属于“服装”类别）。现在，每个标签最多可以出现 λ次，其中 λ 是该方法的一个超参数。作者使用均衡的 k均值聚类初始化划分，然后通过优化方法重新分配标签，以最大化精确率。在Amazon670k基准测试中，该模型达到了最先进水平，$prec@1 = 50.7\%$和 $prec@5 = 41.6\%$。此外，还有采用基于树的搜索的替代模型，能够提高召回率并降低开销 [22]。

极端多标签分类与文本检索有很高的相似性，文本检索将在第6.1节介绍。近年来，这类文本应用取得了长足进展。在稠密检索中，查询和文档表示由BERT模型编码，并返回余弦相似度最高的文档。该领域的许多方法可能也适用于文本分类。

5.1.3 少样本与零样本分类

GPT-2、GPT-3、Gopher 和 PaLM 等大型自回归语言模型通过预训练，已经习得了海量关于事实和语言的信息。正如第 3.6.3 节所述，只需提供少数示例 [76], , 便可以指示它们对文本进行分类。图 5.1 展示了一个根据情感对文本进行分类的提示示例 [91]。这意味着不需要额外的微调数据集，只需提供包含少数示例的提示即可。同样，预训练的 Gopher 模型 [85] 被应用于一组规模庞大的基准任务，约有 150 项，这些任务要求

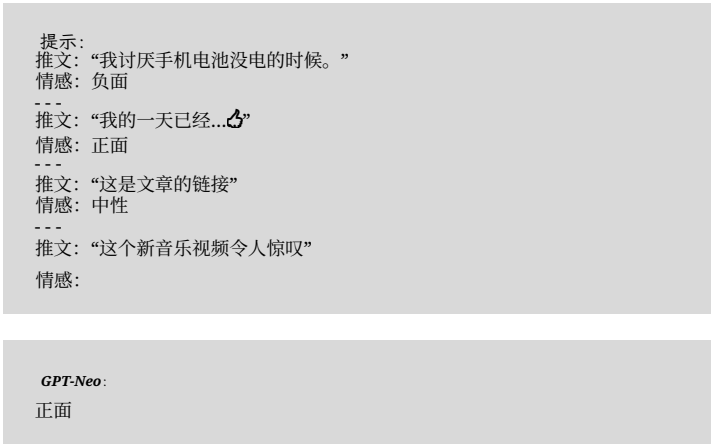


图 5.1 使用 GPT-Neo（一种拥有 27 亿参数的免费版 GPT）进行情感分析少样本学习的查询。该查询可通过 API [91] 进行评估。

使用少样本指令生成答案。与其他自回归模型类似，它可以预测文档的类别标签（第 2.2.5 节）。结果显示[85, 第 56], 页，Gopher 往往能够胜过针对该领域进行微调的传统预训练语言模型。因此，如果能够使用 GPT-3、Gopher 或 GPT-Neo 这类大型自回归预训练语言模型，指令分类似乎是一种可行的替代方案。

最近，*RAFT* [3]基准测试已发布。RAFT 专为评估文本分类任务中的少样本性能而设计。它涵盖 11 个真实世界数据集，其中 8 个是二元分类数据集，两个包含三个类别，一个包含 77 个类别。每项任务都附带自然语言指令和 50 个带标签的训练样本。基准测试示例：“根据句子是否与药物不良反应有关，为句子添加标签。句子：未发现区域性副作用。标签：无关。.....”。提示中包含不到 50 个示例。性能以 11 项任务的平均 F1值衡量。在这些 RAFT 基准测试中，BART 的平均 F1值为 38.2%，GPT-Neo（27 亿参数）达到 48.1%，AdaBoost决策树为 51.4%，GPT-3（1750 亿参数）得分为 62.7%。人类的平均 F1 值为 73.5%。

PET [90] 要求用户指定一个或多个模式，将输入示例x 转换为完形填空提示（第 2.1.2 节），以便像 BERT 这样的掩码语言模型能够处理它。此外，用户必须描述所有输出类别的意义。这通过一个“verbalizer”实现，它会为每个输出 y分配一个自然语言表达。同一数据可以指定多个 verbalizer。例如，对于文本“我真的很喜欢这部电影”，可以使用“我真的很喜欢这部电影。它很 [掩码]。”，也可以使用“我真的很喜欢这部电影。问题：这是一条正面的电影评论吗？回答： [掩码]。”。随后，预训练语言模型会针对观测到的配对进行训练，以最大化 $p(y|x)$。PET 在 RAFT 上达到新的最先进水平，平均 F1值为 82.2%，并且在 11 项任务中的 7 项上表现接近非专家人类。

基础模型也可用于为文本分类任务生成新数据。例如，如果餐厅分类任务需要输入，便可提示模型为特定标签生成一条新的餐厅评论（第 3.6.6 节）。这样就能创建用于微调模型的训练数据。

可用实现

- BigBird、XLNET、T5 等许多经典模型的代码和训练参数可在 Hugging Face https://huggingface.co/transformers/ 获取。
- LightXML 模型的代码位于 https://github.com/kongds/LightXML。
- PET 的代码可在此处获取：https://github.com/timoschick/pet。

5.1.4 总结

对于文档分类，通常会对使用大量文档预训练的预训练语言模型进行微调，以解决特定的分类任务。通常，特定词元（如 [*CLS*] ）的嵌入会作为输入，供逻辑分类器使用。这种设置的表现优于此前所有词袋分类器，例如支持向量机。XLNET 或 ALBERT 等专用预训练语言模型变体表现更佳，因为它们的预训练更为有效。对于较长文档，BigBird 等适用模型能够取得良好结果。识别仇恨言论可视为一项分类任务，使用 BERT 和 RoBERTa 等标准模型即可取得良好结果。

多标签分类的情况则有所不同，因为同一篇文档可能对应多个正确类别。在这种情况下，树形分类器结合上下文嵌入能够取得良好结果。通过树可以选出少量候选类别，从而缩短训练和执行时间。极端多标签分类（例如将产品描述与相关产品描述进行匹配）与文档检索任务相近，因此可以受益于该领域开发的技术，例如使用 DPR 进行稠密检索。

大型预训练自回归语言模型，如 GPT-3、Gopher 和 PaLM，可以通过少样本学习来执行分类任务。近期方法取得的性能已接近人类。不久前发布了一种 API，可用于预训练 GPT-3，并使其适应特定数据和特定分类任务（第 3.6.2 节）。一种更简单的替代方案是 InstructGPT，它可以轻松地被指示执行分类任务，例如情感分析（第 3.6.5 节）。然而，目前尚无对这种方法性能的正式评估，因为模型需要处理训练数据。

虽然预训练语言模型在要求严苛的基准测试中取得了令人鼓舞的结果，但其中大多数模型都无法解释。例如，模型为什么会得出某一特定分类？为什么模型在一个数据集上的表现优于另一个模型，却在其他数据集上的表现更差？尽管注意力机制

和自注意力机制能够对导致特定结果的关联提供一些洞见，但目前仍缺乏对这些模型底层行为和动态的详细研究（第2.4.5节）。深入理解这些模型的理论方面，将有助于人们更好地接受其结果。

5.2 词义消歧

在几乎所有语言中，同一个单词都可能表达不同的概念。例如，单词*“set”*可以是动词、形容词或名词，并可解释为“事物的集合”、“风景”、数学概念、体育术语等。WordNet [62] 词汇数据库列出了该单词的45种不同词义。词义消歧（*WSD*）旨在区分这些不同的意义，并为每个单词标注其词义。它可以被视为一项分类任务，其中每个单词都会被分配到词义清单（如WordNet）中的某个词义。预训练语言模型生成的上下文嵌入提供了一种识别这些意义的方法。Bevilacqua等人 [13]提供了近期的WSD方法综述。

WSD可用于多种目的。一种传统应用是搜索，在搜索查询中区分同一个单词的不同词义。词汇替换Bevilacqua等人 [13] 旨在将文本中的单词或短语替换为另一个意义几乎相同的单词或短语。

5.2.1 义项集合

词义消歧显然取决于义项的定义，而这些义项必须分配给词语。英语词义消歧的主要词义清单是 *WordNet* [62]。它由专家编制的同义词集组成，这些同义词集是表示唯一概念的同义词集合。一个词可以属于多个同义词集，表示其不同的意义。WordNet 3.0版涵盖147,306个词语（或短语）和117,659个同义词集。通过开放多语言*WordNet* 项目，WordNet 也适用于英语以外的其他语言 [17]。维基百科 是另一个常用于实体链接（第5.3.3节）的词义清单，在实体链接中，必须将由维基百科页面表示的人、概念或实体与文本中该实体的给定提及进行链接。*BabelNet*[71]融合了WordNet、维基百科以及其他若干词汇资源，例如维基词典 [111] 和OmegaWiki [75]。它具有高度多语言性，涵盖500多种语言。

WordNet 的词义清单通常过于细粒度。例如，名词 *“star”*在WordNet 中有八种含义。指代“天体”的两种含义，仅区别于该星星是否能从地球上看见。这两种含义在西班牙语中都译为*“estrella”*，因此这种词义区分对这项翻译毫无用处。研究表明，对于许多任务，采用更粗粒度的词义清单效果更好 [81]。

最佳的词义消歧算法使用在大型文档语料库上预训练的预训练语言模型。通过微调，这些模型经过训练，能够从现有词义清单中为词语分配词义。在某些情况下，会采用最近邻操作来衡量嵌入之间的距离，并确定最合适的词义。

5.2.2 模型

GlossBERT [33]采用预训练 BERT 编码器。其微调输入既包括上下文句子（即该词以特定词义使用的句子），也包括词义 *gloss*（定义该词意义的句子）。GlossBERT 经过训练，可预测词义 gloss 是否正确描述了目标词的用法。*SemCor3.0* [61]基准测试采用 WordNet 词义进行标注。GlossBERT 在此数据上取得了新的最先进水平，F1值达到 77.0%。

EWISER [12] 将词义消歧视为简单的词语标注 任务（第 2.1.3 节），为每个单词分配一个词义标签。它首先对来自不同上下文的每个单词的 BERT 嵌入取均值 v_t，然后通过线性层和*Swish* [86] 激活函数 $f(x) = x \cdot \text{sigmoid}(\beta x)$ 对其进行变换。对于每种单词与词性的组合，会确定一个可能词义及其上位词的集合 $S(v_t)$，方法与 [78] 类似。随后，该方法计算单词属于 $S(v_t)$ 中某个同义词集的概率。通过这种方法，预测会考虑某个单词可能对应哪些 WordNet 词义。它在词义消歧基准测试组合上取得了新的最先进水平，达到 80.1%。这个数值也估计了人工标注者间一致性的上限 [69],，表明词义消歧的表现与人类相当。论文列出了多种替代方法的结果。**BEM**模型 [15] 是一个类似的系统，准确率相当。关于预训练语言模型（尤其是 BERT）如何捕捉词汇歧义的详细分析可见 [52]。作者指出，BERT 的嵌入空间包含足够丰富的细节，能够区分不同词义。

MuLaN [9] 基于一个多语言列表 D，其中包含不同语言的同义词集 。例如，D 可能包含与“喷泉”这一“泉水”义项相对应的同义词集，该义项在不同语言中分别表示为“$Quelle_{DE}$”、“$spring_{EN}$”、“$fountain_{EN}$”、“$manantial_{ES}$”、“$brollador_{CAT}$”、“$source_{FR}$”、“$fonte_{IT}$”以及“$sorgente_{IT}$”。使用语义资源库*WordNet* [62]和*BabelNet* [71]来创建 D。MuLaN 的任务是使用源语言（例如英语）中已标注义项的语料库 L_{lab} 作为输入，为目标语言中的未标注语料库 U 标注义项。具体步骤如下：

- 创建嵌入：使用在104种语言上训练的多语言 mBERT（第3.3.1节）计算每个单词 w 在上下文 σ 中、于 Llab里的嵌入$emb(\sigma, w)$。如果 w 被拆分为多个词元，则取这些词元的均值。如果 w是复合词，则先对复合词中每个单词的词元取均值，再对各单词取均值，并将其作为 w的表示。
- 候选项生成：然后，对于上下文 σ 中来自 Llab 的每个嵌入为 w $emb(\sigma, w)$ 的单词，从未标注语料库 U 中找出最近的 1,000 个邻居

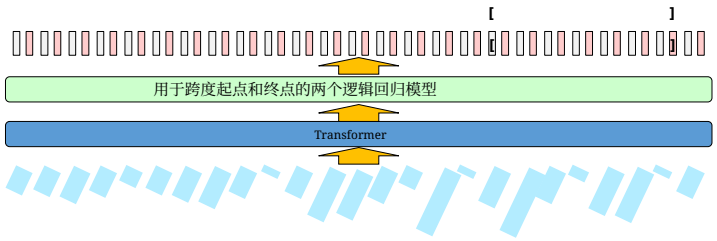


图 5.2 Escher [8] 以一个句子作为输入，其中目标词“背弃”被“<t>”和“</t>”括起来。片段预测所选的句子表示目标词最可能的词义。跨度起点的高概率由“[”表示，跨度终点的高概率由“]”表示。

由 *FAISS* [40]确定。例如，我们从上下文 τ =“*Mihanno consigliato di andarea correre.*”中选取文本跨度 v =“*correre*”，作为句子σ =“*I've seen her go running in thepark.*”中实例 w =“*running*” 的最近候选项 $emb(\tau, v)$。

- 同义词集兼容性：随后，检查最近的候选词 v 是否包含在 D 中 w 的同义词集中。否则将其丢弃。
- 反向兼容性：最后，确定上下文 τ 中 Llab 里的$emb(\tau, v)$ 的最近邻。仅当其最近邻列表包含w 时，才保留 (τ, v)。

- 数据集生成</style}：经过若干额外的筛选步骤后，完成目标语料库中词语的最终标注 U 。

作为有标签语料库 L_{lab} ，使用了带有词义标注的 *SemCor* [63]和*WordNet Gloss Corpus*（*WNG*） [46] 。作为未标注语料库 U ，意大利语、法语、西班牙语和德语使用维基百科。在 *SemEval-13* [70] 和*SemEval-15* [66], 测试时，MuLaN 是这四种语言中表现最好的词义标注系统，F1 值均超过 80%。MuLaN 的一项重要优势是能够将词义标注从高资源语言迁移到低资源语言。

Escher [8] 将词义消歧重新表述为片段预测问题。模型的输入是一个包含目标词及其所有可能词义定义的句子。输出是一个文本片段，用于标识表达目标词最合适意义的词义注释。例如，考虑图 5.2 中的输入句子“<s>霸凌者不得不 <t> 退让 </t>。 </s>”，其中目标词被包围在“<t>”和“</t>”之间。随后附加两个词义注释。

跨度的预测方式与第 2.1.3 节类似，分别计算覆盖正确词义 gloss 的跨度中首个和最后一个词元的概率。在示例中，句子“从某个位置向后移动。”被选作跨度，它描述了正确的词义。通过降低某个单词最常见词义的先验概率，该方法能够减少最常见词义偏见。Escher 使用 $\text{BART}_{\text{LARGE}}$ （第 3.1.3 节）作为预训练语言模型架构，因为它适用于阅读理解。其最后一个解码器层的输出用于表示输入

词元并计算起始和结束词元分布。在多个 SemEval 数据集 [66]上，与竞争方法相比，Escher 的 F1分数更高，而且这一差异在统计学上高度显著。名词和形容词的结果最佳，F1 值为 $> 83\%$，而动词的 F1值仅为 69.3%。

ConSec [10] ConSec 在判断词元的词义时，不仅考虑上下文词，还会考虑分配给相邻词语的词义。它基于 DeBERTa 的扩展版本；DeBERTa 是一种性能更优的 BERT 变体（第 3.1.1 节）。ConSec 使用带有词义注释（词义 gloss）的 WordNet 例句作为额外的训练数据。应用于 *SemCor3.0*基准测试时，该方法的 F1 值达到最先进水平，为 83.2% [61]。

可用实现

- GlossBERT 和 EWISER 的代码及训练模型支持多种语言，可通过以下链接获取：https://github.com/HSLCY/GlossBERT https://github.com/SapienzaNLP/ewiser。
- Escher 及其所需的训练数据可通过以下链接获取：https://github.com/SapienzaNLP/esc。

5.2.3 总结

WSD 可以作为分类任务来处理，即将每个单词分配到若干可能的词义类别中。通常使用 WordNet 作为词义清单。GlossBERT 将单词的上下文嵌入与 WordNet 示例句（词义 gloss ）中单词的嵌入进行比较。EWISER 和 MULAN 直接处理 WordNet 的同义词集，并捕捉可能词义和上位词的集合。它们能够为四种语言标注词义，F1值高于 80%。Escher 将 WSD 重新表述为片段预测问题，使 F1值提高到 83%。ConSec 将邻近词元的词义纳入考量，并取得了相近的性能。

随着 WSD 模型不断改进，人们需要要求更高的基准数据集，这些数据集可能通过对抗技术生成。此外，WSD 模型正趋向于具备更强的领域偏移鲁棒性，并能够处理来自社交媒体文档的文本。为了推动 WSD 的发展，有必要扩展词义标注数据，尤其是针对罕见词义的数据。此外，还可以构建多语言 WSD 系统，这需要大规模的多语言词义消歧基准。WSD 领域正逐渐摒弃固定词义清单，转而以其他方式区分词义，例如执行词汇替换任务，或生成单词在特定上下文中的定义。

一个可能的方向是将词义消歧（第 5.3.3 节）与实体链接整合起来，模型需要将提及与知识库（如维基百科）中的条目关联起来。如今，词义消歧系统的表现已相当不错，因此可以将其与其他应用结合，

例如问答或对话系统。还需要检验，明确纳入词义消歧是否能够带来更好的结果。对于检索任务，词义消歧已被基于嵌入的方法（第 6.1 节）取代，后者能够提供更高的命中率。

5.3 命名实体识别

命名实体识别（*NER*）是指对文本中人物、组织和地点等命名实体的提及进行标签标注的任务。许多领域都存在用于 NER 的标注数据集，例如新闻、科学和医学领域 [72]。通常，这些数据集采用*IOB2* 格式进行标注。例如，该格式使用 B-per 标注人物实体的第一个词元，并使用 I-per 标注该实体的所有其他词元。O 标签用于实体提及之外的所有词元。示例为 “$U.N._{B\text{-}org}$官员$_O$ $Peter_{B\text{-}per}$ $Ekeus_{I\text{-}per}$ 负责$_O$ 前往$_O$ 巴格达$_{B\text{-}loc}$。”NER 涉及对每个词元预测这些标签，即前述示例中的后缀。因此，它可以被视为一项分类任务，为每个词元分配一个标签。用于 NER 的标准数据集是 CoNLL-2003 数据集 [89], ，其中包含带有人物、组织、地点和其他名称标注的英语和德语新闻文本。Li 等人 [48]、Nasar 等人 [68]以及 Bose 等人 [18]提供了关于 NER 的综述。

命名实体识别在专业词汇高度集中的领域尤其有用。例如医疗保健或电动出行领域，每年都会发布数千篇出版物。由于很少有专家能够理解这些术语，命名实体识别系统对于识别专业主题的出版物尤其有价值。当然，命名实体识别类型必须针对每个领域进行调整。

下面一节将介绍普通命名实体识别的方法，其中每个单词只能具有一种实体类型。命名实体也可以嵌套，例如“[[英国]$_{gpe}$驻 [法国]$_{gpe}$]$_{facility}$大使馆]$_{gpe}$”。第二节将讨论这种情况。更具挑战性的是，将一个命名实体短语映射到知识库或本体中的底层唯一实体，例如一个人。这称为实体链接，将在第三节中讨论。

5.3.1 平面命名实体识别

在平面命名实体识别中，每个词元最多对应一个命名实体。**BERT** 可以通过使用逻辑分类器为每个词元预测标签，将其微调用于命名实体识别，逻辑分类器作为最终层（图 2.5）。在此设置下，$\text{BERT}_{\text{LARGE}}$ 在 CoNLL-2003 测试数据上取得了 92.8% 的 F1值。人物实体和地点的 F1 值较高（≈ 95%），而其他名称的 F1值（78%）则低得多，因为这些实体构成了一个定义模糊的类别。

LUKE [117] 将给定文本中的词语和实体视为独立的物体，并输出词元和实体的上下文嵌入。该模型基于 RoBERTa 构建，并在一个从维基百科中提取的大型实体标注语料库上进行训练，以预测随机掩码的词语和实体。通过这种方式，它获得了大量有关文本中实体之间关系的信息。该模型包含一种实体感知自注意力机制，这是 BERT 自注意力机制的扩展，并会考虑嵌入信息，以指示某个词元代表文本还是实体。在 CoNLL-2003 上，它取得了 94.3 的 F1值，接近最先进水平。

ACE [106] 基于这样一个假设：不同词元的嵌入的加权和 $\sum_{i \in I} A_i * \text{嵌入}(v_i)$ 比单个嵌入产生更好的结果。控制器从八种嵌入（如 $\text{BERT}_{\text{BASE}}$、GloVe、FastText 等）中采样一个子集 I，随后训练命名实体识别模型并返回准确率得分。准确率被视为强化学习设置中的奖励信号，并使用策略梯度算法（[112]）选择最优子集 I。命名实体识别模型采用双向长短期记忆网络模型（第1.6节），并以条件随机场层作为最后一层。条件随机场（Conditional Random Field） [100] 能够详细建模标签之间的概率关系。微调模型在 CoNLL-2003 上达到最先进水平，F1值为 94.6%。

KeBioLM [126] 是一种生物医学预训练语言模型，旨在通过纳入额外知识来提升命名实体识别效果。作者从包含生物医学文献摘要的*PubMed* 语料库 [73] 中提取了 6.6 亿个实体，并将其链接到 UMLS 知识库；该知识库包含超过 400 万个实体及其同义词和关系。他们使用 PubMed 数据训练了一个 BERT 变体，并为实体显式生成嵌入。关系信息通过 TransE机制（第3.4.1节）纳入模型。联合损失函数由掩码语言建模、实体检测和实体链接的损失函数构成。*JNLPBA* 基准测试 包含 2,000 篇 PubMed 摘要，其中涉及分子生物学相关实体。KeBioLM 在 JNLPBA 上达到最先进水平，F1值为 82.0%。这表明，在领域文本上进行预训练并纳入额外知识可以提升命名实体识别效果。

检索是一种增强预训练语言模型可用于命名实体识别的上下文的方法。Wang 等人 [107] 使用待标注的输入文本查询搜索引擎。他们根据 RoBERTa 嵌入的相似度对返回结果进行排序（第3.4.5节），并将排名靠前的结果与输入文本拼接起来。随后将其输入 RoBERTa 的一个变体，以生成词元嵌入。由于该模型能够利用对检索文本的注意力，所生成的嵌入可能具有更强的表达能力。CoNLL 2003上的结果表明，检索可将 F1值提高约0.5%，并且可与当前最先进模型结合使用。

5.3.2 嵌套命名实体识别

命名实体通常具有内部结构。此类嵌套实体的一个示例是句子：“昨晚， [[中国]$_{gpe}$ 驻 [法国]$_{gpe}$]$_{facility}$ 大使馆关闭了。”在这种情况下，一个词元可能具有多个实体标签，因此必须以不同方式表述命名实体识别任务。

MRC [50] 将嵌套命名实体识别视为问答任务。例如，提取带有“位置”标签的实体，可以表述为以下问题：“文本中提到了哪些地点？”这些问题使用反映标注指南的模板来表述。针对每种实体类型回答这些问题后，便可检测出重叠命名实体。MRC 使用 BERT 的片段预测方法（第 2.1.3 节），在词元序列中标记某种实体类型跨度的起始位置和结束位置。此外，MRC 还会预测每个实体的起点和终点，以允许同一类型的实体彼此重叠。

嵌套实体在医学领域十分常见。*Genia*语料库 [43]包含蛋白质、病毒、DNA、RNA 等多种实体的标注，其中 17% 的实体为嵌套实体。MRC 在 Genia 基准上取得了 83.8% 的 F1值，达到最先进水平。ACE-2005基准 [104] 包含多种嵌套实体，例如人物、设施或车辆，重叠率为 22%。MRC 在 ACE-2005 上的 F1值达到 86.9%。一种类似的方法 [125] 也能预测不同实体的跨度，在 ACE-2005 上取得了 85.4% 的成绩。Shen 等人提出了一种名为 Locate and Label 的两阶段算法 [93],，该算法先抽取候选实体，再在第二步对其进行分类。使用 BERT 或其变体时，该算法在 ACE-2005 上的嵌套命名实体识别任务中取得了 86.7% 的成绩。

PubMed-BERT 并未使用在通用文档上预训练的 BERT模型，而是仅使用来自 PubMed 的 21GB 医学文本对其拥有 1 亿参数的 BERT模型进行预训练。PubMedBERT 在*BLURB* 基准 [31]的命名实体识别任务中取得了 86.3% 的 F1值。该模型在分类和关系抽取等其他任务上也取得了最先进水平，平均得分为 82.9%。这一结果有力地证明了在领域特定数据上进行预训练的价值。**BioELECTRA**[42] 是一种生物医学领域特定语言编码器模型，它将 ELECTRA（第 3.1.1 节）应用于生物医学领域。ELECTRA 采用一种样本效率高的“替换词元检测”技术进行预训练，使模型能够从训练数据中获取海量信息。BioELECTRA 使用 PubMed 和 PubMed Central 中的医学文章全文进行预训练。在命名实体识别任务中，它在 BLURB 基准 [31]上取得了最高得分，F1值为 86.7%。在其他 BLURB 任务上，该模型也取得了与 PubMedBERT 相近的得分，为 82.6%。

可用实现

- BERT大型 词元分类 https://huggingface.co/transformers/model_doc/model_doc/ bert.html，
- Lukehttps://huggingface.co/transformers/model_doc/model_doc/luke.html
- ACEhttps://github.com/Alibaba-NLP/ACE，
- MRChttps://github.com/ShannonAI/mrc-for-flat-nested-ner
- 定位并标注 [93] https://github.com/tricktreat/locate-and-label
- 用于嵌套命名实体识别的 Bioelectra https://github.com/kamalkraj/BioELECTRA

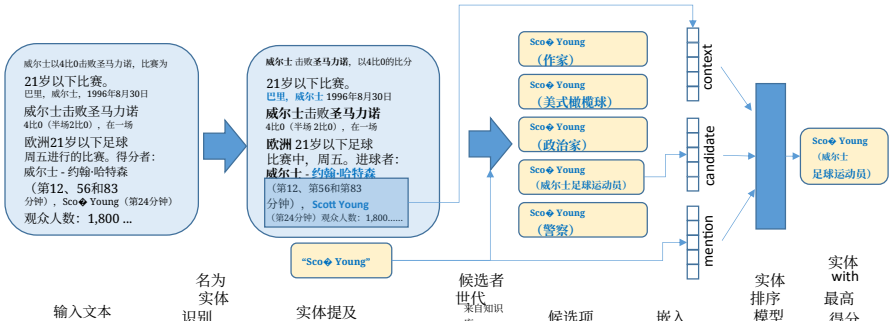


图 5.3 实体链接包含三个步骤：实体识别，用于识别文本中的实体提及；候选生成，利用知识库为提及生成可能的实体；以及实体排序，计算候选项与提及之间的相似度分数。图像改编自 [92],，经作者许可转载。

5.3.3 实体链接

在文本中识别出命名实体（实体提及）后，人们通常希望对其进行消歧，即将该提及关联到知识库或本体中的唯一实体。这涉及统一实体名称的不同写法。为了将相应的事实和关系关联到同一实体，链接名称的不同写法十分重要，例如 *“Joe Biden* 当选为美国第*46*任总统*”* 以及 *“*拜登总统出生于宾夕法尼亚州斯克兰顿*”*。请注意，名称 *“Muammar Muhammad Abu Minyar al-Gaddafi”*大约有35种写法，例如 *“Qadhafi”*、*“Gaddafi”* 和*“Gadhafi”*，此外还有名字部分不同的写法。实体链接 方法旨在解决这一问题。

实体链接对于知识库构建、聊天机器人、推荐系统和问答等任务非常有用，可用于识别所指的正确宾语或实体。对于需要实体身份的模型，例如 KnowBERT [80] 或 ERNIE [99] （第 3.4.1 节），实体链接也是预处理步骤。早期方法依赖语义嵌入来匹配彼此相关的实体提及 [82]。现代方法使用上下文嵌入来表征实体提及。Sevgili 等人 [92] 对基于深度学习的实体链接方法进行了全面综述。他们概述了实体链接方法的一般解决方案架构，如图 5.3 所示，并比较了不同的方法。

BLINK [113] 遵循图 5.3 所示的方法。首先，通过命名实体识别从文本中提取实体提及及其类型。然后，使用 BERT 模型计算提及上下文和知识库中实体描述的嵌入。这也涉及实体名称的规范化。使用高效的近似 k最近邻索引方案 FAISS [40] 对嵌入进行索引（第 6.1.4 节）。FAISS 能够以较低的计算成本，从知识库中检索最匹配的实体候选项。这种方法与 DPR 的稠密检索相同（第 3.4.5 节）。每个

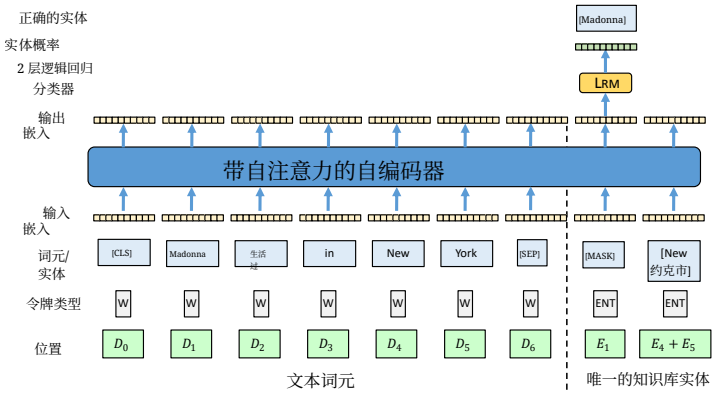


图 5.4 $\text{BERT}_{\text{LARGE}}$ 可以通过微调来预测掩码后的“实体标记”，同时考虑相应的文本。应用过程中，会依次分配概率最高的实体。这样便可以利用实体的联合概率 [118]。

检索到的候选项随后会通过交叉编码器进行更仔细的检查。交叉编码器将输入上下文、提及和实体文本拼接起来，并为每个候选实体分配一个得分。最后，选取得分最高的候选项。尽管该方法没有显式计算实体嵌入，但在*TACKBP-2010* 基准 [29] 上达到了最先进水平，准确率为 94.5%。**EntQA** [130], 也采用了非常类似的方法，同样利用检索器-阅读器架构，并在多个基准测试上取得了有竞争力的结果。

GENRE [25]不同于大多数实体链接方法采用的常见解决方案架构，而是使用编码器-解码器模型 BART（第 3.1.3 节）对实体进行消歧。该模型在预训练期间需要恢复由多种不同方法破坏的文本，因此积累了大量语言知识。模型经过微调后，可生成消歧后的命名实体。例如，句子 *“In*1503*, Leonardobegan painting theMona Lisa.”* 被转换为*“In* 1503*,* [*Leonardo*](*Leonardo da Vinci*) *began painting the* [*Mona Lisa*](*MonaLisa*)*.”*，其中“[*Leonardo*](*Leonardo da Vinci*)”和“*Mona Lisa*”是维基百科中对应条目的唯一标题。GENRE 使用受约束的束搜索进行解码，可以复制输入文本，也可以生成唯一的维基百科实体名称。此外，GENRE 还能执行提及检测和端到端实体链接，将提及关联到对应的知识库实体（例如维基百科条目）。在六个不同的基准测试中，GENRE 的平均 F1值达到 88.8%，优于得分为 77.0% 的 BLINK。此外，GENRE 的内存占用（2.1GB）小于 BLINK（30.1GB）。最后，该模型倾向于原样复制提及，这有助于处理新的、未见过的命名实体。

EntMask [118] 与 LUKE（第 3.4.4 节）类似，会学习预测掩码实体。为了消歧新的提及，作者使用基于词语的局部上下文信息，以及基于已完成消歧的

实体。该模型经过训练，能够联合生成词语和实体的嵌入，并且同样基于 $\text{BERT}_{\text{LARGE}}$。微调时，会随机掩盖 30% 与维基百科超链接相对应的实体，并按图 5.4 所示对其进行预测。应用过程中，模型会为每个提及预测一个实体，并将尚未解决的提及中概率最高的那个实际指定为“已观察到”。如此一来，这一指定便会影响对剩余提及的预测，从而引入全局视角。在若干基准测试中，该方法取得的结果与 GENRE 大致相当，并在少数基准测试中略占优势。

可用的实现

- GENRE：来自 Facebook 的模型源代码和数据集

https://github.com/facebookresearch/ GENRE
- BLINK 可在以下位置获取： https://github.com/facebookresearch/BLINK
- EntMask 代码： https://github.com/studio-ousia/luke。

5.3.4 总结

众所周知，命名实体在理解文本含义方面发挥着至关重要的作用。每天都有成千上万个新的命名实体出现，因此需要付出特别的努力来理解其含义。由于预训练语言模型（PLM）中上下文嵌入的可用性，命名实体识别（NER）能够将 CoNLL 2003 基准测试上的 F1值从 85% 提高到 94.6%，大幅减少错误。标准方法是使用 BERT 进行词元标注，为每个词元标记相应的实体类型。通过将命名实体视为特殊词元（LUKE）、结合不同类型的嵌入（ACE），或使用基于嵌入的检索方法，可以获得更高的性能。实证评估表明，在领域文本（例如来自医学领域的文本）上训练底层预训练语言模型至关重要。单个词元或复合词可以同时属于多个实体类型。为此，可以采用嵌套式命名实体识别问答方法，将词元跨度标记为属于某一实体类型。同样，在领域文本上进行训练也至关重要。

第 5.4.4 节介绍了实体与关系联合抽取的方法。该节描述的方法也可单独用于命名实体识别，并有望取得较高的性能。REBEL 就是一个示例，它使用 BART 编码器-解码器将输入句子转换为所涵盖实体和关系的唯一表示。

实体链接旨在将实体提及映射到知识库中的相应唯一实体。一种方法利用检索器-阅读器架构从知识库中查找实体候选项（BLINK、EntQA）。随后，阅读器模块会仔细审查候选项和该提及，以确定最终分配结果。另一种方

表格 5.3 基于关系抽取的语言分析任务 [4, 页 10]

Task	描述	示例
共指消解	将指向同一宾语的短语分组。	*Betty*(1) 喜欢她的(1) 可爱狗狗(2)。
基于方面的情感分析	从文本中提取短语（方面）并确定它们的情感（正面、负面、中性）。	这块牛排$_{aspect}$ 真是糟透了$_{negative}$。
实体关系抽取	抽取实体或文本中的概念之间的关系。	*Peter* 是一名律师。 → 职业*(Peter,* 律师*)*
事件抽取	抽取事件，即 n 元关系文本中的实体或名词之间。	中午$_{time}$ 恐怖分子$_{attacker}$ 引爆了一枚炸弹$_{instrument}$ 在巴黎$_{place}$。 →冲突-攻击
语义角色标注	确定每个动词所对应的短语相对于该动词的角色。	*Mary*$_{agent}$ 出售了$_{verb}$ 这本书$_{theme}$ 给 *John*$_{recipient}$。

法是 GENRE 的编码器-解码器架构，它将实体提及转换为唯一实体名称。最后，BERT模型可以确定词元嵌入与实体嵌入之间的自注意力，并利用该信息预测文本中包含的唯一实体。

大多数实体链接模型在候选生成步骤中仍依赖于维基百科等外部知识。然而，在识别非名人的人物实体时，这并不充分。在这种情况下，我们必须在 Web 或社交媒体中进行搜索以查找信息。随着检索-阅读器方法日益普及，这在未来或许能够实现。事实证明，命名实体识别和实体链接应当联合执行，即各种指派应相互考虑，以提高准确率。

5.4 关系抽取

识别出句子中的相关实体后，信息抽取的关键环节通常是抽取这些实体之间的关系并对其进行分类。例如，当我们希望自动将带有链接的信息填入数据库或知识图谱时，这项工作就很有用。表 5.3 列出了本节讨论的基于关系抽取的语言分析任务示例。其中包括共指消解，即在同一文本中找出指向同一实体的不同提及；基于方面的情感分析，即将文本中的短语与针对这些短语的观点关联起来；以及语义角色标注，即识别短语在句子中对于谓词所起的作用。实体链接将实体提及与本体中对应的唯一底层对象或人物关联起来，因此不同于关系抽取。Nasar 等人 [68] 对关系抽取的相关既有研究进行了综述。

5.4.1 共指消解

关系抽取的第一种类型是共指消解，其目标是建立文本中所有指向同一现实世界实体的实体提及之间的关系。例如，考虑句子“我投票支持拜登，因为他的立场与我的价值观最为一致”，她说道。其中“我”、“我的”和“她”指的是扬声器，而“拜登”和“他”指的是 Joe Biden。由于相关短语子集的组合数量庞大，共指分析是自然语言处理领域最具挑战性的任务之一。Stylianou 等人对共指消解进行了综述。 [98]

SpanBERT [41] 是 BERT 的一个版本，在预训练期间预测掩码词元的连续子序列，因此积累了关于词语跨度的知识（第 3.1.1 节）。作者考虑文本的所有可能跨度，并识别相关的提及跨度。同时，对于每个跨度 x，都会检查其前置跨度 y，并通过评分函数估计这些跨度是否指向同一实体。

该评分函数定义为 $s(x,y) = s_m(x) + s_m(y) + s_c(x,y)$。其中，$s_m(x)$ 和 $s_m(y)$ 用于衡量 x 和 y 是实体提及的可能性。$s_c(x,y)$ 用于确定 x 和 y 指向同一实体的可能性。评分函数以一个跨度作为输入，接收该跨度两个端点的输出嵌入，以及该跨度词元嵌入的总结。y 与 x 共指的概率计算为 $p(y) = \exp(s(x,y))/\sum_{y' \in Y} \exp(s(x,y'))$。通过这种方式，可以形成提及同一实体的跨度子集。在该方法的迭代过程中，跨度定义可能会得到细化，同时应用先行词剪枝机制，以减少需要考虑的跨度数量。OntoNotes [109] 是一个包含 150 万词的语料库，涵盖多种体裁的文本，并包含共指等结构信息。在 OntoNotes 上进行微调后，Span-BERT 在测试集上取得了 79.6% F1值 的最先进水平结果。Dobrovolskii [27] 提出了一种变体，在词级进行分析，从而降低任务的复杂度。该方法将 OntoNotes 上的最先进水平提升至 81.0%。

CorefQA [114] 将共指消解作为问答问题来解决。第一阶段将所有不超过最大长度的跨度视为潜在提及。作者使用aSpanBERT模型计算所有词元的嵌入。为减少提及数量，结合跨度起始和结束嵌入的提议模块经过预训练，以预测相关提及。随后，每个提及依次被特殊词元包围，并训练网络标记所有与之共指的跨度，其方式类似于BERT的问答微调（第2.1.3节）。为减少计算量，只考虑一个方向上的有限数量候选项。提及提议和提及聚类可以进行端到端训练。在共指基准测试CoNLL 2012 [84]上，该方法将最先进水平显著提升至83.1%的F1值。Toshniwal 等人 [103] 通过每次仅跟踪少量有界实体来扩展这一方法。即使对于长文档，该方法在共指消解中也能达到高准确率。

可用实现

- GitHub上的SpanBERT关系抽取与共指消解 https://github.com/facebookresearch/SpanBERT
- GitHub 上的 CorefQA https://github.com/ShannonAI/CorefQA

5.4.2 句子级关系抽取

可以抽取多种类型的关系，例如，在句子“歌德在魏玛因病逝世”中，“逝世于”关系将一个人物（“歌德”）与一个位置（“魏玛”）关联起来。在本节中，我们假设实体已通过命名实体识别（第 5.3 节）从句子中抽取。因此，命名实体识别错误会增加关系抽取中的错误。

SpanBERT [41] 特别适用于关系抽取，因为实体提及通常跨越多个词元，并且在预训练期间会被 SpanBERT 掩码（第 3.1.1 节）。微调时，模型接收一个句子和两个包含可能关系论元的跨度作为输入，并将它们替换为相应的命名实体识别标签。示例：“[*CLS*] [*SUBJ-PER*] 出生于 [*OBJ-LOC*]，密歇根州，..... ”。最终的 [CLS] 嵌入作为逻辑分类器的输入，分类器会从 42 种预定义关系类型（包括“无关系”）中预测一种。*Re-TACRED* [97]是一个大规模关系抽取数据集，包含 12 万个示例，涵盖 41 种关系类型（例如 per:schools-attended 和 org:members），其关系标注经过仔细核验。SpanBERT 在 Re-TACRED 上表现良好，F1值为 85.3% [95]。

RoBERTa （第 3.1.1 节）可用于生成关系抽取的词元嵌入。Zhou 等人 [135] 评估了各种实体表示技术。他们通过最后一层的嵌入，使用 RoBERTa $_{\text{LARGE}}$ 对输入文本进行编码。每个关系论元提及跨度中第一个词元的嵌入被用于表示这些论元。将这些嵌入连接起来，并将其作为softmax分类器的输入。结果表明，在 Re-TACRED 数据集上，使用特殊词元包围实体并添加其类型可以获得最佳结果，F1值达到 91.1%。

Relation-QA [24] 将关系分类问题重新表述为问答问题。考虑句子 s =“*Sam Brown* 于 *1991* 年出生。”以及抽取出的实体 “*Sam Brown*” 和 “1991”。随后，作者创建两个查询，例如 “*Sam Brown* 是什么时候出生的？” 和 “谁出生于 *1991* 年？”。他们微调 ALBERT（第 3.1.1 节），通过标记包含目标实体的跨度来回答这些查询。如果没有返回任何跨度，则该关系不成立。该方法在 TACRED 上取得了 74.8% 的 F1值；TACRED 是 ReTACRED 的较早版本，存在许多标注问题。 **RECENT** [55] 扩展了 SpanBERT，并训练多个关系分类模型，即针对每种不同的实体类型对分别训练一个分类器。这限制了可能的输出关系类型，有助于提升性能。该方法在 TACRED 上取得了 75.2% 的最先进水平 F1值。

5.4.3 文档级关系抽取

尤其是对于较大的文档，认为关系只出现在句子内部这一假设过于严格。因此，一些模型会在文档层面检测关系。当关系论元位于不同句子中时，相应实体通常只通过共指提及来指代。因此，本节假设实体已经被抽取，并通过共指消解归入表示同一实体的簇中（第 5.4.1 节）。显然，共指消解产生的错误会增加最终关系抽取中的错误。

SSAN [115]（结构化自注意力网络）直接将共指和实体提及共现等结构信息纳入考量，适用于 RoBERTa 等预训练语言模型。作者通过添加特定偏见，修改编码器块中的自注意力计算：当两个提及指向同一实体和/或位于同一句子中时，便添加这些偏见。这些偏见通过查询向量和键向量计算得出，所用的“变换模型”是在微调过程中训练的。因此，键与查询之间的标量积会根据对应词元是否共指、是否位于同一句子中而改变。实体嵌入通过对实体提及的词元嵌入进行平均池化得到。对于每一对嵌入$_i$，嵌入$_j$，通过双线性变换 sigmoid(嵌入$_i$ W_r 嵌入 $_j$) 计算其关系 r 的概率，变换使用可训练的参数矩阵W_r。 T

DocRED [121] 是一个大型文档基准测试，其中标注了命名实体、共指和关系，这些关系的论元可能位于不同的句子中。作者以 $\text{RoBERTa}_{\text{LARGE}}$ 作为基础网络，在 DocRED 上取得了 65.9% 的 F1值，达到最先进水平。使用在 Semantic Scholar 的科学论文上训练的特殊 BERT 版本 SciBERT [11] 后，该算法在包含化学和生物学文本的基准测试中也取得了最先进水平的结果。

ATLOP [136] 通过特殊词元标记提及的开始和结束位置，并使用 BERT 对文档进行编码，从而为每个词元生成嵌入。提及起始位置词元的嵌入被用作提及嵌入。实体嵌入通过对共指提及进行池化计算得到。关系中第一个和第二个论元实体的嵌入分别经过不同的全连接层变换为 $\boldsymbol{x}_1$和 $\boldsymbol{x}_2$。随后，通过稀疏双线性变换 sigmoid($\boldsymbol{x}_1$ $W\boldsymbol{x}_2$)估计实体对之间关系 r 的概率。使用可训练的概率阈值来判断是否存在某种关系。在 DocRED 基准上，该模型取得了 63.4% 的 F1值。 T

5.4.4 实体与关系联合抽取

由于命名实体识别和关系抽取密切相关，且关系抽取依赖于命名实体识别的结果，因此将这两项任务联合建模是自然而然的选择。

UniRE [108] 在一个联合矩阵中编码实体和关系的性质，该矩阵为每个文本词元设置一行和一列。命名实体（例如 PER）标记在对角线上

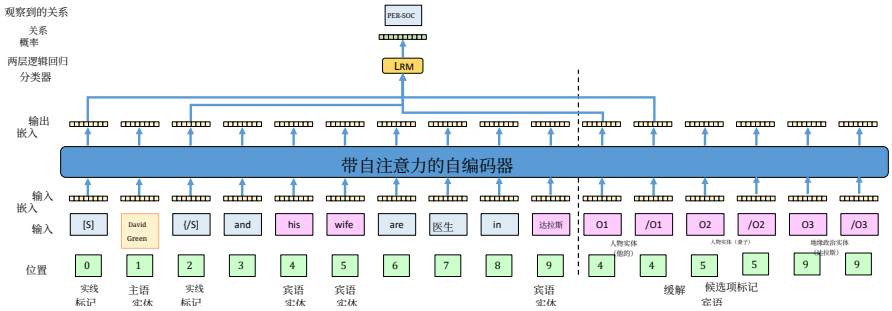


图 5.5 对于一种可能的关系，PL 标记模型使用特殊的“实心”标记标出第一个关系论元，并在文本之外使用“弱化”标记标出可能的第二个论元。后者与对应词元的位置相同，并且在注意力计算期间不会影响普通词元的嵌入。标记嵌入会拼接起来，以计算对应关系的概率 [122]。

，关系则是矩阵中的非对角线元素。例如，如果*“DavidPerkins”* 住在“加利福尼亚州”，则*“David Perkins”*词元所在行与“加利福尼亚州”词元所在列的矩阵元素会标记为 *PHYS*关系。请注意，这种方式可以表示非对称关系。

输入中的所有单词都使用 BERT 编码器进行编码，然后使用双仿射模型为一对嵌入 h_i 和 h_j 生成评分向量

$$p(y_{i,j}|s) = \text{softmax}\left((\boldsymbol{h}_i^{first})^\intercal U_1 \boldsymbol{h}_j^{sec} + U_2[\boldsymbol{h}_i^{first}, \boldsymbol{h}_j^{sec}] + b\right), \tag{5.2}$$

其中，$\boldsymbol{h}_i^{first} = \text{Fcl}^{first}(\boldsymbol{h}_i)$ 和 $\boldsymbol{h}_i^{sec} = \text{Fcl}^{sec}(\boldsymbol{h}_i)$ 分别是对第一个和第二个关系论元进行变换的全连接层。softmax 函数会为所有矩阵单元格中的实体标签和关系标签生成概率分布。该模型会最小化三个损失函数：一个基于各单元格的实际标签，一个基于实体标签的对角线应当对称这一知识，另一个则基于这样一个事实：关系标签意味着相应的实体标签必须存在。*ACE 2005* [104] 包含多种类型的文本，并对其中的实体、关系和事件进行了标注。在 ACE 2005 上，UniRE 的实体与关系联合抽取 F1值为 66.0%，低于当前 70.5% 的最先进水平。

PL-Marker [122]研究不同类型的提及编码。对于可能的关系，它使用实心标记词元包围第一个论元跨度（主语）。可能的第二个论元跨度（宾语）则由抬升标记 *Oi*以及 */Oi* 文本外部的标记进行标记（图 5.5）。这些标记与文本中相应宾语跨度具有相同的位置嵌入。它们的注意力连接受到限制，即彼此可见，但对文本词元以及其他成对的标记不可见。因此，根据主语跨度的不同，宾语词元嵌入可以捕捉不同的方面。对于每一对主宾论元，将相应的嵌入进行拼接，并将其用作逻辑分类器的输入，以估计

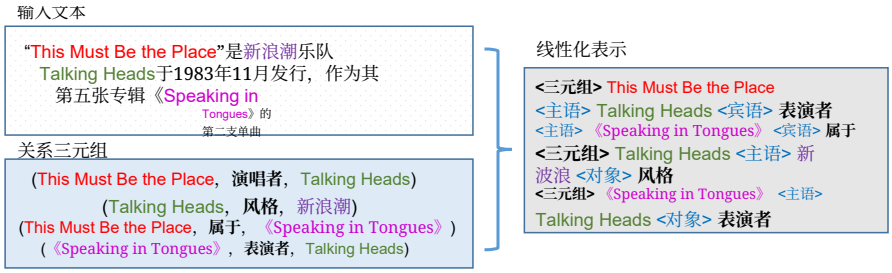


图 5.6 对于训练集，左侧的关系信息被线性化为右侧的表示。因此，REBEL 模型学习将输入文本转换为这种线性化表示 [20]。

可能关系（或“无关系”）的概率。使用 ACE 2005 对 BERT 的预训练变体进行微调，以预测这些关系。采用具有 1.05 亿个参数的 $\text{BERT}_{\text{BASE}}$模型时，该方法在 ACE05 基准上的 F1值达到 68.8%。如果使用具有 2.35 亿个参数的 $\text{ALBERT}_{\text{XXLARGE}}$ [45]来计算嵌入，F1值将提高到 72.3%。

对于命名实体识别，PL-Marker 模型采用了类似的方法。对于输入中每个可能的跨度（从词元 v_i开始，到词元 $v_{j,j\geq i}$结束），都会创建抬高标记，这些标记不会影响普通词元的嵌入。同样，跨度起始和结束词元的嵌入以及抬高标记的嵌入都会输入逻辑分类器，以计算不同命名实体类型的概率。该模型采用高效的“打包”方式来降低计算开销。在 CoNLL03 命名实体基准测试上，使用预训练 $\text{RoBERTa}_{\text{LARGE}}$ 的 PL 标记模型取得了 94.0 的 F1值，远低于 DeBERTa[19]目前保持的最先进水平 96.1%。当关系抽取使用 PL-MARKER 命名实体识别所预测的实体类型和跨度时，联合方法的 F1值降至 70.5%，这一成绩是 ACE05 基准上联合命名实体识别与关系抽取的最先进水平。

REBEL [20] 使用编码器-解码器Transformer $\text{BART}_{\text{LARGE}}$ （第 3.1.3 节）进行实体与关系联合抽取，输出输入文本中出现的每个关系（h, r, t）三元组。它将包含实体及其隐含关系的原始输入句子转换为一组三元组，明确表示这些关系。示例见图 5.6。文本中的每个关系都会按照其第一个参数的位置出现在输出中。一个实体可能属于不同的关系，这些关系按照第二个参数的位置排序。这就确定了线性化表示中关系的顺序。

拥有 4 亿个参数的预训练 $\text{BART}_{\text{LARGE}}$ 首先在包含 220 种关系类型的维基百科和维基数据训练集上进行微调。随后，它在不同的基准数据集上进行第二次微调。在 *DocRED* 基准 [121] 上，其 F 值达到 47.1%，达到最先进水平。在《纽约时报》数据集 上，其性能达到最先进水平，F1值为 93.4%。在 ReTACRED 基准上，即使不使用其他方法所采用的实体类型标记，它的 F1值仍可达到 90.4%。

基于方面的情感分析

基于方面的情感分析也称为方面级情感分析、基于特征的情感分析，或简称方面情感分析，可帮助组织对成员或客户反馈数据进行详细分析。其应用范围从分析客户对餐厅的反应，到评估人们对政治家发表的政治声明的态度。例如："服务员$_{1\text{-}aspect}$非常友好$_{1\text{-}positive}$，但菲力牛排$_{2\text{-}aspect}$烤得焦煳$_{2\text{-}negative}$。"请注意，一个句子可能包含不同的方面，因此必须将每种情感分配给某一个方面。Zhang 等人对基于方面的情感分析进行了近期综述。 [129]

DeBERTa （第 3.1.1 节）是一种强大的类 BERT 模型，假设方面已知。它采用解耦注意力机制，在词语和位置之间分别计算注意力分数，从而解耦文本数据的语义（内容）表示和句法（位置）表示。其目标是确定给定实体的每个方面的情感。输入由一段文本和一个方面组成，例如 $x =$"[*CLS*] ... *nice video camera and keyboard...* [*SEP*] *keyboard* [*SEP*]"，其中 *"keyboard"* 是文本 [94]中的一个可能的方面跨度。 [*CLS*] 的输出嵌入被用作逻辑分类器的输入，该分类器生成三个可能标签（正面、负面、中性）的概率。该模型在 *SemEval 2014* 任务 *4.2*基准上进行微调。在餐厅和笔记本电脑数据上的平均准确率达到 86.1%。还有更复杂的方法，例如**LSA** （局部情感聚合） [119]，在该基准测试上达到了 88.6%的最先进水平。

GRACE [54] 旨在同时抽取方面和标签。它由一个首层 $\text{BERT}_{\text{BASE}}$ 模块组成，该模块生成输入文本的词元嵌入，并通过微调为每个词元标注 IOB2 标签，以标记方面。随后将生成的信息输入 Transformer 解码器，为每个词元预测情感（正面、负面、中性）。该解码器使用多头交叉注意力，以纳入首个方面模块提供的信息。同样，对于最后一层中的每个词元嵌入，都会使用逻辑分类器来计算各种情感的概率。为提高模型的鲁棒性，训练期间会对输入词元嵌入施加微小扰动。请注意，由于解码器不是自回归的，因此不需要掩码交叉注意力。这样，模型在标注情感时便能够考虑方面术语之间的交互。在 SemEval 2014 的笔记本电脑评论数据集上，该模型的方面抽取 F1分数达到 87.9%；在方面与情感的联合抽取任务上，F1值达到最先进水平 70.7%。在餐厅评论数据集上，其 F1 分数为 78.1% ；在推文基准数据集上，联合情感抽取的 F1分数为 58.3%，再次优于其他若干模型。

语义角色标注

语义角色标注会考察句子中的谓词（例如动词），并根据短语的句法角色对其进行分类，例如施事、球门或结果。

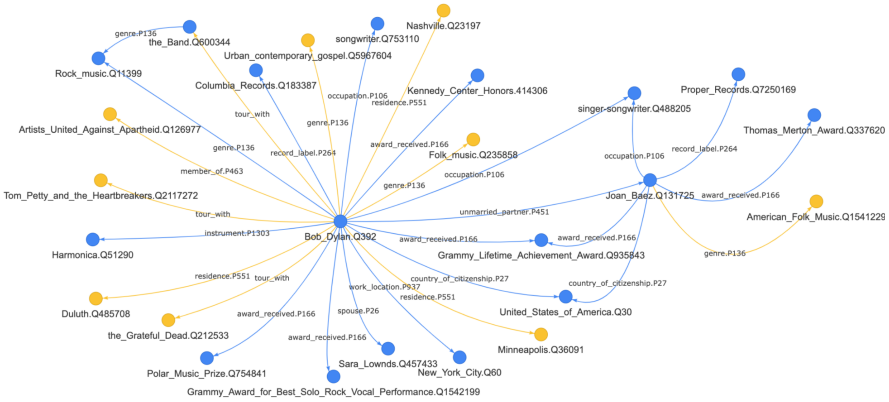


图 5.7 使用 MAMA [105]，借助 $\text{BERT}_{\text{LARGE}}$，从邻近 *"Bob Dylan"* 的维基百科页面生成的开放知识图谱的一个快照子图。蓝色节点和箭头表示 Wikidata 模式中已映射的事实，而黄色节点和箭头表示开放模式中尚未映射的事实。Wikidata 中新增的正确事实以黄色显示。图像来源： [105, 第 6], 页，经作者许可使用。

它可用于确定句子的意义。例如，考虑句子"他们想做得更多。"，其中"想"是谓词，"他们"是施事，而"做得更多"是宾语（想要做的事情）。

Crf2o [133]是一种树结构条件随机场（treecrf） [28]，以 RoBERTa 计算的输入词元上下文嵌入作为输入。输入序列 $\boldsymbol{x} = (x_1, \ldots\ldots, x_T)$ 可以排列成一棵树 $\boldsymbol{y}$ 并获得一个得分，该得分是其所有子树得分之和 $s(\boldsymbol{x}, \boldsymbol{y}) = \sum_{t \in \boldsymbol{y}} s(\boldsymbol{x}, t)$。与依存句法分析类似，这可以用于建模语义角色标注中短语相对于谓词的依存关系 [87]。生成所有可能的子树需要进行 T^3 次运算，效率非常低。作者利用结构约束成功降低了这一开销。此外，他们还可以考虑树的两个分支之间的依存关系，由此生成二阶树。在训练过程中，模型会最大化对于给定输入，训练数据所提供树结构的概率。*CoNLL05* [21] 和 *OntoNotes* [84] 是语义角色标注中广泛使用的两个基准测试。对于 CoNLL05，Crf2o 的 F1值为 89.6%；对于 OntoNotes，其 F1值达到 88.3 %，二者均达到新的最先进水平。请注意，这项技术也可用于依存句法分析 [132],，该任务通过树结构描述句子的句法结构。

从预训练语言模型中提取知识图谱

Wang 等人 [105] 展示了一种从大型语言模型中提取知识的系统方法。他们的 **MaMa** 方法由匹配阶段和

映射阶段组成。匹配阶段利用语言模型的内部知识，从文本集合中生成一组候选事实。与 TransE（第 3.4.1 节）类似，每个事实都表示为一个关系三元组（注意力头、关系、尾巴），或（h, r, t）。语言模型用于生成与 r 或 t 相对应的词元。作为条件，r 值应为连续的文本序列，并表达常见关系。

在映射阶段，三元组会被映射为具有适当关系的相关三元组。例如，根据 Wikidata 模式，(Dylan, is, songwriter) 会被映射为 (Bob Dylan.Q392, occupation.P106, Songwriter.Q753110)。这一阶段与第 5.3.3 节讨论的实体链接有关。映射到现有知识图谱模式的原因，是为了利用专家设计的高质量模式。

图 5.7 展示了生成的关系子图。与最先进水平的信息抽取系统 Stanford OpenIE [5]（F1 值为 27.1%）相比，该方法的 F1 值达到 29.7%。作者报告称，性能会随着模型规模增大而提升，因为更大的模型能够存储更多知识。

可用实现

- PL-Marker 代码和模型可在此公开获取：https://github.com/thunlp/PL-Marker。
- REBEL 在 GitHub 上的页面 https://github.com/babelscape/rebel 和 Hugging Face 上的页面 https://huggingface. co/Babelscape/rebel-large
- MaMa：源代码和预训练模型见https://github.com/theblackcat102/language-models-are-knowledge-graphs-pytorch

5.4.5 远程监督

获取用于关系抽取的大型标注数据集是一项繁琐的任务，而且往往会因隐私问题而难以完成。由于大量关系知识存储在知识库中，Mintz 等人 [65] 提出了远程监督范式。其核心思想是收集所有两个实体共同出现的文本提及，而这两个实体在知识库中存在某种关系。然后假定这对提及表达了该关系。由于并非所有这类提及对都符合这一假设，许多方法致力于消除这种“噪声”。其中一种方法是多示例学习，它放宽了原先“所有文本提及对都表示该关系”的假设，改为假定至少有一对提及 [2, 137], 或某个指定比例（例如 10%）的提及能够表达该关系，具体比例也可取决于得分值。以实体“巴拉克·奥巴马”和“夏威夷”为例，它们在知识库中可能存在“出生_于”这一关系。通过搜索这两个实体的出现位置获得的句子可能包括“奥巴马出生于夏威夷”以及“奥巴马在夏威夷度家庭假期”，其中只有前者表达了该关系，应当用于训练。

KGPool [67] 使用从知识库中获取的实体对，同时也利用与其相关的属性。其思路是创建实体节点、句子以及

知识库中实体节点属性的表示，包括描述、实例-类别关系和别名属性。所有这些信息均通过单词和字符嵌入以及双向长短期记忆网络进行嵌入，并连接成一个异构信息图。接下来，使用三层图卷积网络和读出层。通过对读出表示应用自注意力，计算 Softmax 分数，再依据分数和一个超参数进行筛选，从而只选取相关属性节点。随后创建动态掩码，池化掉不太重要的实体属性节点。最后，将实体、句子和读出的所有中间表示分别拼接，形成最终的实体、句子和读出表示。然后将这些表示与关系表示一起输入全连接层，并通过 Softmax 激活函数计算每种关系的得分。《纽约时报》数据集 是采用远程监督进行关系抽取的标准基准测试。KGPool 达到最先进水平，其 Precision@10 为 92.3%；该指标表示在匹配结果中选取排名最靠前的 10 个结果时，相关结果所占的比例。

5.4.6 利用布局信息进行关系抽取

要理解正式文本，通常除了文本内容外，还必须考虑文档布局。尤其是在表单类文本中，词语的位置和填写的值都很重要。在第 7.2 节中，我们将介绍如何通过一个或多个 Transformer 同时处理文本和图像，从这两种媒体中提取意义。作为铺垫，我们将利用 Transformer 处理多模态输入的能力，并通过二维位置特征纳入布局信息。 [96] 对版面分析的进展进行了全面综述。本小节将聚焦于键值抽取方法。键值抽取任务旨在分析文档，从中提取与关注的书面键对应的印刷值。典型应用包括发票的自动处理，其中的键是属性，例如发票日期或应付总金额。

ReLIE [57]是一种用于从表单类文档中抽取键值的框架。候选生成步骤旨在为特定键找到所有可能的值候选项，例如，键“日期”对应的值“1/16/2018”。这些值候选项通常对应数字、金额、日期等基本类型，可通过基于规则的匹配器找到。随后训练一个基于 Transformer 的评分模型，以从抽取出的值候选项中识别有效值。为此，模型会学习键、值候选项的位置以及邻近词元及其位置的嵌入。值候选项及其每个邻居的位置通过各自边界框质心的二维笛卡尔坐标来描述。请注意，为避免过拟合，候选值的文本不会进行编码。自编码器中的自注意力机制将所有嵌入相互关联。然后，通过余弦相似度比较字段嵌入和候选项嵌入，并将所得得分缩放到 $[0, 1]$ 的范围内。该

模型在发票键值抽取任务上的 F1值 达到 87.8%，在收据上的 F1值 达到 83.3%。

DocFormer [6] 由一个 CNN 视觉骨干网络和一个仅含编码器的 Transformer 架构组成。文档的视觉嵌入由 ResNet50 模型生成，并通过线性层投影到合适的嵌入维度。文本词元位于边界框内，每个词元边界框的左上角和右下角位置分别通过两个不同的矩阵转换为嵌入。此外，还会对相邻边界框之间的高度、宽度和距离进行编码。二维位置嵌入通过一维位置嵌入融入绝对位置。视觉特征和文本特征分别使用独立的空间嵌入进行训练。DocFormer 的注意力机制是对原始注意力机制的改进版本。针对词元的视觉表示和文本表示，分别计算注意力分数。除了键—查询注意力外，还使用查询词元和键词元的相对位置嵌入，为视觉嵌入和文本嵌入分别加入相对位置注意力和空间注意力。视觉表示和文本表示共享空间注意力权重。

DocFormer通过三种不同的预训练任务进行预训练：多模态掩码语言建模（MM-MLM）、学习重建（LTR）和文本描述图像（TDI）。在MM-MLM任务中，词元会被掩码，模型需要将其重建。在LTR中，给定文档的多模态表示，模型需要重建文档图像。使用平滑L1损失计算原始图像与重建图像之间的差异。TDI需要执行文本-图像匹配任务，模型必须针对随机样本预测图像和文本是否匹配。*FUNSD* 基准 [38]涵盖199份扫描文档中的表单；如果词元属于同一个键，就必须将它们分组。在FUNSD基准上，DocFormer的F1值达到84.6%，在发表时处于最先进水平。

LayoutLM3 [34] 采用了一种受视觉Transformer启发的图像嵌入方法（见第7.2.2节）。每幅图像被划分为 16×16 个图像块，与视觉Transformer中的做法类似，并通过线性变换转换为嵌入。如图5.8所示，词语和图像块由同一个自回归Transformer处理。预训练时，模型使用掩码语言建模任务、掩码图像块任务和词-图像块对齐预训练任务。在掩码图像块任务中，模型必须重建图像块。词-图像块对齐任务旨在使模型学会对齐文本和视觉表示。模型应判断词元的文本与图像块是否对齐，即两者均未被掩码，或未对齐，即图像块被掩码。*PubLayNet* 基准 [134] 包含超过100万份PDF文档的布局，并与正确的文档结构相匹配。在此基准上，LayoutLM3以边界框平均精度均值94.5%的成绩达到最先进水平。它在FUNSD键值抽取任务和其他基准测试中的表现优于DocFormer。*LayoutXLM* 是LayoutLM3 [116]近期推出的多语言版本。

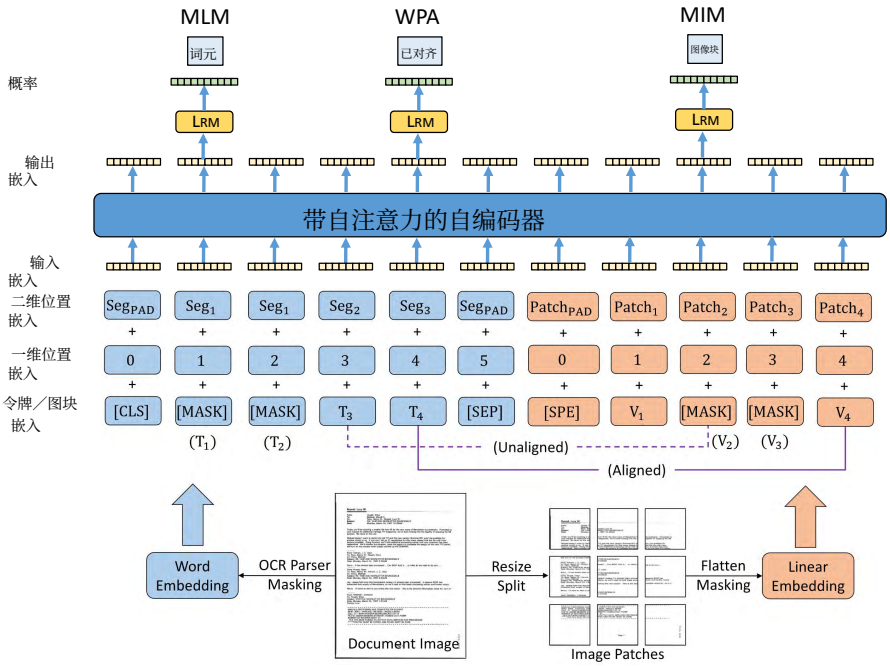


图 5.8 LayoutLMv3 将图像块和词元的线性投影作为输入，并将其编码为上下文化的向量表示。LayoutLMv3 使用掩码语言建模（MLM）和掩码图像建模（MIM）的离散词元重构目标进行预训练。此外，LayoutLMv3 还通过词-图像块对齐（WPA）目标进行预训练，以预测文本单词对应的图像块是否被掩码，从而学习跨模态对齐。“Seg”表示片段级位置。图像来源： [34, 第 3], 页，经作者慷慨许可转载。

可用实现

- KGPool 位于 https://github.com/nadgeri14/KGPool

5.4.7 总结

关系抽取的任务是评估文本中针对特定实体所表达的关系。例如，客户对某些产品特征的评估有助于改进产品或服务。面对海量文本内容，人工处理观点信息已难以实现。

对于简单情况，关系论元是已知的，因此可以使用 RoBERTa、DeBERTa 或 SpanBERT 等 BERT 变体，将关系抽取作为简单的分类任务来解决。然而，要实际使用这些模型，我们必须先提取关系论元，这会导致总体误差增加。

更具挑战性的是同时抽取关系论元及相应的关系类型，因为这些任务相互依赖。UniRE 在联合矩阵中标注实体和关系，并在自注意力计算中引入相应的偏见。PL-Marker 使用特殊词元标记第一个关系论元，并使用所谓的抬升标记标记第二个论元。这些词元具有特定的注意力性质，能够提升模型在流行基准测试上的性能。GRACE 采用特定的编码器-解码器架构，其中编码器标注关系论元（方面），解码器则为每个词元分配关系标签。REBEL 使用 BART 编码器-解码器，将输入句子转换为所涵盖关系的唯一表示。

关系抽取模型已经针对特定应用进行了调整。GRACE 针对基于方面的情感分析进行了调优，而 Crf2o 则针对语义角色标注进行了调优。后者使用上下文嵌入，并通过高效的 TreeCRF 确定谓词与相应短语之间的关系。最后，MaMa 可用于根据实体之间抽取的关系构建知识图谱。

文档和网页的空间布局通常包含有助于抽取关系论元的相关信息。在这种情况下，可以利用文档图像中的视觉信息，得出有效的解读。视觉信息可以通过键和值的边界框位置纳入，也可以通过图像块的形式纳入；后者将在后文结合 Image Transformer 进行探讨。

近期所有关系抽取方法都基于预训练语言模型。大多数模型在实验中使用小型 BERT变体。因此，可以假设更大的模型会直接提升性能。此外，GPT-3 等基础模型可以进行微调（第 3.6.2 节），并且可能会带来更高的准确率。另一种相关的替代方案是 InstructGPT（第 3.6.5 节），它可以轻松地通过问答引导执行关系抽取，例如 “谁建造了自由女神像？” [77, 第 29] 页。不过，似乎很难根据某些测试数据评估这种方法的性能。

参考文献

[1] J. Abreu、L. Fred、D. Macêdo 和 C. Zanchettin。《用于文档分类的分层注意力混合神经网络》。载于：国际人工神经网络会议。施普林格，2019年，第396–402页。[2] L. Adilova、S. Giesselbach 和 S. Rüping。《高效利用领域专家在关系抽取中的时间》。2018年。arXiv：1807.04687。[3] N. Alex 等。《RAFT：真实世界少样本文本分类基准》。1月2022 2109 14076 [ ] 18; 。

arXiv：。cs。[4] Z. Alyafeai、M. S. AlShaibani 和 I. Ahmad。《自然语言处理中的迁移学习综述》。2020年。arXiv：2007.04239。

[5] G. Angeli、M. J. J. Premkumar 和 C. D. Manning。《利用语言结构进行开放领域信息抽取》。载于：计算语言学协会第53届年度会议论文集及自然语言处理第7届国际联合会议论文集。第1卷，长篇论文。2015年，第344–354页。[6] S. Appalaraju、B. Jasani、B. U. Kota、Y. Xie 和 R. Manmatha。《DocFormer：用于文档理解的端到端Transformer》。载于：IEEE/CVF计算机视觉国际会议论文集。2021年，第993–1003页。[7] S. Auer、C. Bizer、G. Kobilarov、J. Lehmann、R. Cyganiak 和 Z. Ives。《DBpedia：开放数据网络的核心》。载于：《语义网》。K. Aberer 等编。计算机科学讲义。柏林、海德堡：施普林格，2007年，第722–735页。isbn：978- 3-540-76298-0。doi：10.1007/978-3-540-76298-0 52。_[8] E. Barba、T. Pasini 和 R. Navigli。《ESC：利用抽取式词义理解重新设计词义消歧》。载于：计算语言学协会北美分会2021年会议人类语言技术论文集。2021年，第4661–4672页。[9] E. Barba、L. Procopio、N. Campolungo、T. Pasini 和 R. Navigli。《MuLaN：用于词义消歧的多语言标签传播》。载于：IJCAI论文集。2020年，第3837–3844页。[10] E. Barba、L. Procopio 和 R. Navigli。《ConSeC：将词义消歧视为连续词义理解》。载于：2021年自然语言处理经验方法会议论文集。2021年，第1492–1503页。[11] I. Beltagy、K. Lo 和 A. Cohan。《SciBERT：用于科学文本的预训练语言模型》。2019年。arXiv：1903.10676。[12] M. Bevilacqua 和 R. Navigli。《突破80%的玻璃天花板：通过融入知识图谱信息提升词义消歧的最先进水平》。载于：计算语言学协会论文集。2020年，第2854–2864页。[13] M. Bevilacqua、T. Pasini、A. Raganato 和 R. Navigli。《词义消歧的近期趋势：综述》。载于：第三十届人工智能国际联合会议论文集。IJCAI-21。国际人工智能联合会议公司，2021年。[14] K. Bhatia、K. Dahiya、H. Jain、P. Kar、A. Mittal、Y. Prabhu 和 M. Varma。《极端分类资源库》。2021年6月7日。url：http://manikvarma.org/downloads/XC/XMLRepository.html（访问于2021年6月7日）。[15] T. Blevins 和 L. Zettlemoyer。《沿着词义消歧的长尾继续前进：利用释义信息引导的双编码器》。2020年。arXiv：2005.02590。[16] P. Bojanowski。fastText。2016年。url：https://fasttext.cc/index.html（访问于2021年2月21日）。[17] F. Bond 和 R. Foster。《链接并扩展开放多语言WordNet》。载于：计算语言学协会第51届年度会议论文集。第1卷，长篇论文。2013年，第1352–1362页。[18] P. Bose、S. Srinivasan、W. C. Sleeman、J. Palta、R. Kapoor 和 P. Ghosh。《临床文本上的近期命名实体识别与关系抽取技术综述》。载于：应用科学。11.18（2021年），第8319页。[19] F. Brandon。Brandon25/Deberta-Base-Finetuned-Ner - Hugging Face。2021年10月12日。url：https://huggingface.co/brandon25/deberta-base-finetuned-ner（访问于2022年2月15日）。[20] P.-L. H. Cabot 和 R. Navigli。《REBEL：通过端到端语言生成进行关系抽取》。载于：计算语言学协会发现：EMNLP 2021。2021年，第2370–2381页。[21] X. Carreras 和 L. Màrquez。《CoNLL-2005共享任务简介：语义角色标注》。载于：第九届计算自然语言学习会议论文集。CoNLL-2005。2005年，第152–164页。[22] W.-C. Chang 等。《用于商品搜索语义匹配的极端多标签学习》。2021年6月23日。arXiv：2106.12657 [cs]。[23] G. Choi、S. Oh 和 H. Kim。《利用句子重要性改进文档级情感分类》。载于：熵。22.12（2020年），第1336页。[24] A. D. Cohen、S. Rosenman 和 Y. Goldberg。《作为双向跨度预测的关系抽取》。2020年。arXiv：2010.04829。[25] N. De Cao、G. Izacard、S. Riedel 和 F. Petroni。《自回归实体检索》。2021年3月24日。arXiv：2010.00904。

[26] S. Ding、J. Shang、S. Wang、Y. Sun、H. Tian、H. Wu 和 H. Wang。《ERNIE-DOC：回顾式长文档建模 Transformer》。2020年。arXiv：2012.15688。[27] V. Dobrovolskii。《词级共指消解》。2021年。arXiv：2109.04127。[28] J. Eisner。《双词法文法及其立方时间句法分析算法》。载于：《概率句法分析及其他句法分析技术进展》。施普林格，2000年，第 29–61 页。[29] D. Gillick、S. Kulkarni、L. Lansing、A. Presta、J. Baldridge、E. Ie 和 D. Garcia-Olano。《用于实体检索的稠密表示学习》。2019年。arXiv：1909.10506。[30] GitHub。GitHub。2021年。网址：https://github.com/。[31] Gu。BLURB排行榜。2021年。网址：https : / / microsoft . github . io / BLURB/（访问于2022年02月13日）。[32] J. He、L. Wang、L. Liu、J. Feng 和 H. Wu。《通过循环注意力学习从局部词语片段进行长文档分类》。载于：《IEEE Access》7（2019年），第 40707–40718 页。[33] L. Huang、C. Sun、X. Qiu 和 X. Huang。《GlossBERT：利用释义知识进行词义消歧的 BERT》。2019年。arXiv：1908.07245。[34] Y. Huang、T. Lv、L. Cui、Y. Lu 和 F. Wei。《LayoutLMv3：采用统一文本与图像掩码的文档人工智能预训练》。2022年。arXiv：2204.08387。[35] huggingface。Transformers——Transformers 4.3.0 文档。2021年。网址：https : / / huggingface.co/transformers/（访问于 2021年02月21日）。[36] M. S. Jahan 和 M. Oussalah。《使用自然语言处理进行仇恨言论自动检测的系统综述》。2021年。arXiv：2106.00742。[37] K. Jasinska、K. Dembczynski、R. Busa-Fekete、K. Pfannschmidt、T. Klerx 和 E. Huller-meier。《使用稀疏概率估计实现极端 F值最大化》。载于：《国际机器学习会议》。PMLR，2016年，第 1435–1444 页。[38] G. Jaume、H. K. Ekenel 和 J.-P. Thiran。《Funsd：噪声扫描文档中的表单理解数据集》。载于：2019 年国际文档分析与识别会议研讨会 ICDARW。第 2 卷。IEEE，2019年，第 1–6 页。[39] T. Jiang、D. Wang、L. Sun、H. Yang、Z. Zhao 和 F. Zhuang。《Lightxml：采用动态负采样实现高性能极端多标签文本分类的 Transformer》。2021年。arXiv：2101.03305。[40] J. Johnson、M. Douze 和 H. Jégou。《千亿级相似性搜索与图形处理器》。载于：《IEEE 大数据汇刊》（2019年）。[41] M. Joshi、D. Chen、Y. Liu、D. S. Weld、L. Zettlemoyer 和 O. Levy。《SpanBERT：通过表示和预测跨度改进预训练》。载于：《计算语言学协会会刊》8（2020年），第 64–77 页。[42] K. raj Kanakarajan、B. Kundumani 和 M. Sankarasubbu。《BioELECTRA：使用判别器的预训练生物医学文本编码器》。载于：第 20 届生物医学语言处理研讨会。BioNLP-NAACL 2021。线上：计算语言学协会，2021年6月，第 143–154 页。doi：10.18653/v1/2021.bionlp-1.16。[43] J.-D. Kim、T. Ohta、Y. Tateisi 和 J. Tsujii。《GENIA语料库——用于生物文本挖掘的语义标注语料库》。载于：《生物信息学》19（增刊_1 2003），第 i180–i182 页。[44] K. Kowsari、K. Jafari Meimandi、M. Heidarysafa、S. Mendu、L. Barnes 和 D. Brown。《文本分类算法：综述》。载于：《Information》10.4（2019年），第 150 页。[45] Z. Lan、M. Chen、S. Goodman、K. Gimpel、P. Sharma 和 R. Soricut。《ALBERT：用于语言表示自监督学习的轻量级 BERT》。2020年。arXiv：1909. 11942。[46] H. Langone、B. R. Haskell 和 G. A. Miller。《标注 WordNet》。新泽西州普林斯顿：普林斯顿大学认知科学实验室，2004年。[47] Q. V. Le 和 T. Mikolov。《句子和文档的分布式表示》。2014年5月22日。arXiv：1405。4053 [cs]。[48] J. Li、A. Sun、J. Han 和 C. Li。《基于深度学习的命名实体识别综述》。载于：《IEEE 知识与数据工程汇刊》（2020年）。[49] Q. Li 等人。《文本分类综述：从浅层学习到深度学习》。2020年。arXiv：2008.00364。

[50] X. Li、J. Feng、Y. Meng、Q. Han、F. Wu 和 J. Li。《用于命名实体识别的统一机器阅读理解框架》。2019年。arXiv：1910.11476。[51] X. Liu、W.-C. Chang、H.-F. Yu、C.-J. Hsieh 和 I. S. Dhillon。《基于分区的极端多标签分类中的标签解耦》。2021年。arXiv：2106.12751。[52] D. Loureiro、K. Rezaee、M. T. Pilehvar 和 J. Camacho-Collados。《语言模型用于词义消歧的分析与评估》。载于：《计算语言学》2021年，第 47 卷第 2 期，第 387–443 页，2021年3月。[53] E. Loza Mencía 和 J. Fürnkranz。《法律领域大规模问题的高效成对多标签分类》。载于：《联合欧洲机器学习与知识发现数据库会议》。施普林格，2008年，第 50–65 页。[54] H. Luo、L. Ji、T. Li、N. Duan 和 D. Jiang。《GRACE：用于基于方面的情感分析的梯度协调级联标注》。2020年。arXiv：2009.10557。[55] S. Lyu 和 H. Chen。《带实体类型限制的关系分类》。2021年。arXiv：2105.08393。[56] A. Maas、R. E. Daly、P. T. Pham、D. Huang、A. Y. Ng 和 C. Potts。《用于情感分析的词向量学习》。载于：第 49 届计算语言学协会人类语言技术年会论文集，2011年，第 142–150 页。[57] B. P. Majumder、N. Potti、S. Tata、J. B. Wendt、Q. Zhao 和 M. Najork。《从表单类文档中进行信息抽取的表示学习》。载于：第 58 届计算语言学协会年会论文集，2020年，第 6495–6504 页。[58] T. Mandl、S. Modha、P. Majumder、D. Patel、M. Dave、C. Mandlia 和 A. Patel。《FIRE 2019 HASOC赛道概览：印欧语言中的仇恨言论与冒犯性内容识别》。载于：第 11 届信息检索评估论坛。FIRE '19：信息检索评估论坛。印度加尔各答：ACM，2019年12月12日，第 14–17 页。isbn：978-1-4503-7750-8。doi：10.1145/3368567.3368584。[59] B. Mathew、P. Saha、S. M. Yimam、C. Biemann、P. Goyal 和 A. Mukherjee。《HateXplain：用于可解释的仇恨言论检测的基准数据集》。2021年。arXiv：2012 10289。cs。[60] [ ] J. McAuley 和 J. Leskovec。《隐含因子与隐含主题：利用评论文本理解评分维度》。载于：第 7 届 ACM 推荐系统会议论文集，2013年，第 165–172 页。[61] R. Mihalcea。SemCor语料库。2008年6月13日。网址：https://kaggle.com/nltkdata/semcor-corpus（访问于2022年01月04日）。[62] G. A. Miller。《WordNet：英语词汇数据库》。载于：《ACM通讯》38.11（1995年），第 39–41 页。[63] G. A. Miller、C. Leacock、R. Tengi 和 R. T. Bunker。《语义索引》。载于：人类语言技术会议论文集：1993年3月21日至24日在新泽西州普莱恩斯伯勒举行的研讨会。1993年。[64] S. Minaee、N. Kalchbrenner、E. Cambria、N. Nikzad、M. Chenaghlu 和 J. Gao。《基于深度学习的文本分类：全面综述》。载于：《ACM计算综述》CSUR 54.3（2021年），第 1–40 页。[65] M. Mintz、S. Bills、R. Snow 和 D. Jurafsky。《无标注数据的远程监督关系抽取》。载于：第 47 届 ACL 年会与第 4 届自然语言处理联合国际会议 AFNLP 联合会议论文集，2009年，第 1003–1011 页。[66] A. Moro 和 R. Navigli。《SemEval-2015任务13：多语言全词词义消歧与实体链接》。载于：第 9 届语义评测国际研讨会论文集。SemEval 2015。2015年，第 288–297 页。[67] A. Nadgeri、A. Bastos、K. Singh、I. O. Mulang、J. Hoffart、S. Shekarpour 和 V. Saraswat。《KGPool：用于关系抽取的动态知识图谱上下文选择》。2021年。arXiv：2106.00459。[68] Z. Nasar、S. W. Jaffry 和 M. K. Malik。《命名实体识别与关系抽取：最先进水平》。载于：《ACM计算综述》CSUR 54.1（2021年），第 1–39 页。[69] R. Navigli。《词义消歧：综述》。载于：《ACM计算综述》CSUR 41.2（2009年），第 1–69 页。

[70] R. Navigli、D. Jurgens 和 D. Vannella。《SemEval-2013 任务12：多语言词义消歧》。载于：第二届词汇与计算语义联合会议（SEM），第2卷：第七届语义评估国际研讨会论文集（SemEval 2013）。2013年，第222–231页。[71] R. Navigli 和 S. P. Ponzetto。《BabelNet：广覆盖多语言语义网络的自动构建、评估与应用》。载于：人工智能，193（2012年），第217–250页。[72] ner。Papers with Code——命名实体识别。2021年。网址：https://paperswithcode.com/ task/named-entity-recognition-ner（访问于2021年7月9日）。[73] 美国国立卫生研究院。下载数据。PubMed。2022年。网址：https://pubmed.ncbi.nlm.nih.gov/download/（访问于2022年6月15日）。[74] 自然语言处理。自然语言处理索引。2021年。网址：https://index.quantumstat.com/。[75] Omegawiki。OmegaWiki。2021年。网址：http://www.omegawiki.org/（访问于2022年1月3日）。[76] OpenAi。OpenAI API。2021年。网址：https://beta.openai.com（访问于2021年11月14日）。[77] L. Ouyang 等。《训练语言模型遵循指令并获得人工反馈》。2022年1月31日。arXiv：2203.02155。[78] G. Paaß 和 F. Reichartz。《利用语义约束，通过条件随机场估算上位语义类别》。载于：2009年 SIAM 国际数据挖掘会议论文集。SIAM，2009年，第485–496页。[79] Papers-with-code。Papers with Code。2021年。网址：https://paperswithcode.com/。[80] M. E. Peters、M. Neumann、R. L. Logan IV、R. Schwartz、V. Joshi、S. Singh 和 N. A. Smith。《知识增强的上下文词表示》。2019年。arXiv：1909.04164。[81] M. T. Pilehvar、J. Camacho-Collados、R. Navigli 和 N. Collier。《将词义无缝融入下游自然语言处理应用》。2017年。arXiv：1710.06632。[82] A. Pilz 和 G. Paaß。《利用主题语境距离从姓名识别实体》。载于：第20届 ACM 国际信息与知识管理会议论文集。2011年，第857–866页。[83] Y. Prabhu、A. Kag、S. Harsola、R. Agrawal 和 M. Varma。《Parabel：用于极端分类的分区标签树及其在动态搜索广告中的应用》。载于：2018年万维网国际会议论文集。2018年，第993–1002页。[84] S. Pradhan、A. Moschitti、N. Xue、O. Uryupina 和 Y. Zhang。《CoNLL-2012 共享任务：在 OntoNotes 中对多语言非受限共指进行建模》。载于：EMNLP-CoNLL 共享任务联合会议。2012年，第1–40页。[85] J. W. Rae 等。《扩展语言模型：Gopher 训练的方法、分析与见解》。载于：ArXiv 预印本 ArXiv211211446（2021年12月8日），第118页。[86] P. Ramachandran、B. Zoph 和 Q. V. Le。《搜索激活函数》。2017年。arXiv：1710.05941。[87] F. Reichartz、H. Korte 和 G. Paass。《基于类型化依存树核函数的语义关系抽取》。载于：第16届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集。2010年，第773–782页。[88] S. G. Roy、U. Narayan、T. Raha、Z. Abid 和 V. Varma。《利用多语言 Transformer 检测仇恨言论》。2021年。arXiv：2101.03207。[89] E. F. Sang 和 F. De Meulder。《CoNLL-2003 共享任务简介：语言无关的命名实体识别》。2003年。arXiv：cs/0306050。[90] T. Schick 和 H. Schütze。《借助提示实现真正的少样本学习——现实世界视角》。2021年11月26日。arXiv：2111.13440 [cs]。[91] P. Schmid。实践中的少样本学习：GPT-Neo 与加速推理API。2021年6月3日。网址：https://huggingface.co/blog/few-shot-learning-gpt-neo-and-inference-api（访问于2022年5月23日）。[92] O. Sevgili、A. Shelmanov、M. Arkhipov、A. Panchenko 和 C. Biemann。《神经实体链接：基于深度学习的模型综述》。2020年。arXiv：2006.00575。[93] Y. Shen、X. Ma、Z. Tan、S. Zhang、W. Wang 和 W. Lu。《定位并标注：用于嵌套命名实体识别的两阶段识别器》。2021年。arXiv：2105.06804。[94] E. H. Silva 和 R. M. Marcacini。《使用解耦注意力的 BERT 基于方面的情感分析》。载于：（2021年）。网址：https://repositorio.usp.br/bitstreams/701d2a63-e3f4- 450d-8617-ad80de4345ed。

[95] Spanbert。Papers with Code——机器学习最新进展。2021年7月17日。网址：https://paperswithcode.com/paper/spanbert- improving- pre- training- by/review/?hl=28781（访问于2021年7月17日）。[96] T. Stanisławek。文档理解精选资源。2022年7月2日。网址：https://github.com/ tstanislawek/awesome-document-understanding（访问于2022年7月8日）。[97] G. Stoica、E. A. Platanios 和 B. Póczos。《Re-Tacred：弥补 Tacred 数据集的不足》。载于：AAAI 人工智能会议论文集，第35卷，第15期。2021年，第13843–13850页。[98] N. Stylianou 和 I. Vlahavas。《神经实体共指消解综述》。载于：Expert Systems with Applications，168（2021年），第114466页。[99] Y. Sun 等。《ERNIE：通过知识融合增强表示》。2019年。arXiv：1904.09223。[100] C. Sutton 和 A. McCallum。《条件随机场与关系学习导论》。载于：统计关系学习导论，2（2006年），第93–128页。[101] T. Thongtan 和 T. Phienthrakul。《使用通过余弦相似度训练的文档嵌入进行情感分类》。载于：第57届计算语言学协会年会学生研究研讨会论文集。意大利佛罗伦萨：计算语言学协会，2019年7月，第407–414页。doi：10.18653/v1/P19-2057。[102] R. Tinn 等。《面向生物医学自然语言处理的大型神经语言模型微调》。2021年12月14日。arXiv：2112.07869 [cs]。[103] S. Toshniwal、S. Wiseman、A. Ettinger、K. Livescu 和 K. Gimpel。《学会忽略：使用有界记忆神经网络处理长文档共指》。2020年。arXiv：2010.02807。[104] C. Walker、S. Strassel、J. Medero 和 K. Maeda。ACE 2005 多语言训练语料库。语言数据联盟，2006年2月15日。doi：10.35111/MWXC-VH88。[105] C. Wang、X. Liu 和 D. Song。《语言模型是开放的知识图谱》。2020年10月22日 2010 11967 。arXiv：。。[106] X. Wang、Y. Jiang、N. Bach、T. Wang、Z. Huang、F. Huang 和 K. Tu。《用于结构化预测的嵌入自动拼接》。2020年。arXiv：2010.05006。[107] X. Wang、Y. Jiang、N. Bach、T. Wang、Z. Huang、F. Huang 和 K. Tu。《通过外部上下文检索和协同学习提升命名实体识别》。2021年。arXiv：2105.03654。[108] Y. Wang、C. Sun、Y. Wu、H. Zhou、L. Li 和 J. Yan。《UniRE：实体关系抽取的统一标签空间》。2021年。arXiv：2107.04292。[109] R. Weischedel、M. Palmer、R. B. S. P. L. Ramshaw、N. Xue 和 E. Hovy。《OntoNotes：用于增强处理的大型训练语料库》。载于：Joseph Olive、Caitlin Christ 和 John McCary 编，《自然语言与机器翻译手册》，DARPA 全球语言探索项目（2011年）。[110] G. Wiedemann、S. M. Yimam 和 C. Biemann。《UHH-LT 在 SemEval-2020 任务12中的表现：微调预训练 Transformer 网络以检测攻击性语言》。2020年6月10日。arXiv：2004.11493 [cs]。[111] wiktionary。维基词典。2021年。网址：https://www.wiktionary.org/（访问于2022年1月3日）。[112] R. J. Williams。《用于联结主义强化学习的简单统计梯度跟随算法》。载于：Machine Learning，8.3（1992年），第229–256页。[113] L. Wu、F. Petroni、M. Josifoski、S. Riedel 和 L. Zettlemoyer。《利用稠密实体检索实现可扩展的零样本实体链接》。载于：2020年自然语言处理实证方法会议论文集（EMNLP）。2020年，第6397–6407页。[114] W. Wu、F. Wang、A. Yuan、F. Wu 和 J. Li。《将共指消解视为基于查询的跨度预测》。2020年7月18日。arXiv：1911.01746。[115] B. Xu、Q. Wang、Y. Lyu、Y. Zhu 和 Z. Mao。《实体结构的内外建模：对文档级关系抽取中的提及依赖关系进行建模》。2021年。arXiv：2102.10249。[116] Y. Xu 等。《LayoutXLM：面向多语言视觉丰富文档理解的多模态预训练》。2021年。arXiv：2104.08836。

[117] I. Yamada、A. Asai、H. Shindo、H. Takeda 和 Y. Matsumoto。“LUKE：通过实体感知自注意力实现深度上下文化实体表示”。2020年。arXiv：2010.01057。[118] I. Yamada、K. Washio、H. Shindo 和 Y. Matsumoto。“利用词语和实体的预训练上下文化嵌入进行全局实体消歧”。2021年11月24日。arXiv：1909.00426 [cs].[119] H. Yang、B. Zeng、M. Xu 和 T. Wang。“回归现实：利用模式驱动建模，实现经济高效的情感依存学习”。2021年。arXiv：2110. 08604。[120] Z. Yang、Z. Dai、Y. Yang、J. Carbonell、R. R. Salakhutdinov 和 Q. V. Le。“XLNet：用于语言理解的通用自回归预训练”。载于：神经信息处理系统进展。2019年，第5753–5763页。[121] Y. Yao 等。“DocRED：大型文档级关系抽取数据集”。2019年。arXiv：1906.06127。[122] D. Ye、Y. Lin 和 M. Sun。“打包处理：借助悬浮标记进行实体与关系抽取”。2021年。arXiv：2109.06067。[123] W. Yin 和 A. Zubiaga。“迈向可泛化的仇恨言论检测：障碍与解决方案综述”。载于：PeerJ Comput. Sci. 7（2021年），e598。[124] R. You、Z. Zhang、Z. Wang、S. Dai、H. Mamitsuka 和 S. Zhu。“AttentionXML：一种基于标签树、具备注意力机制的深度模型，用于高性能极端多标签文本分类”。2018年。arXiv：1811.01727。[125] J. Yu、B. Bohnet 和 M. Poesio。“将命名实体识别视为依存句法分析”。2020年。arXiv：2005.07150。[126] Z. Yuan、Y. Liu、C. Tan、S. Huang 和 F. Huang。“利用知识改进生物医学预训练语言模型”。2021年。arXiv：2104.10344。[127] M. Zaheer 等。“Big Bird：面向更长序列的 Transformer”。载于：神经信息处理系统进展 33（2021年1月8日）。[128] M. Zampieri 等。“SemEval-2020 任务12：社交媒体中的多语言冒犯性语言识别（OffensEval 2020）”。2020年。arXiv：2006.07235。[129] W. Zhang、X. Li、Y. Deng、L. Bing 和 W. Lam。《基于方面的情感分析综述：任务、方法与挑战》。2022年3月2日。doi：10.48550/arXiv.2203.01054。arXiv：2203.01054 [cs].[130] W. Zhang、W. Hua 和 K. Stratos。“EntQA：将实体链接视为问答”。2021年。arXiv：2110.02369。[131] X. Zhang、J. Zhao 和 Y. LeCun。“用于文本分类的字符级卷积网络”。2015年。arXiv：1509.01626。[132] Y. Zhang、Z. Li 和 M. Zhang。“用于神经依存句法分析的高效二阶 TreeCRF”。2020年。arXiv：2005.00975。[133] Y. Zhang、Q. Xia、S. Zhou、Y. Jiang、Z. Li、G. Fu 和 M. Zhang。“将语义角色标注视为依存句法分析：探索论元内部的潜在树结构”。2021年。arXiv：2110.06865。[134] X. Zhong、J. Tang 和 A. J. Yepes。《PubLayNet：有史以来规模最大文档版面分析数据集》。2019年8月15日。doi：10.48550/arXiv.1908.07836。arXiv：1908.07836 [cs]。[135] W. Zhou 和 M. Chen。“句子级关系抽取的改进基线”。2021年。arXiv：2102.01373。[136] W. Zhou、K. Huang、T. Ma 和 J. Huang。“结合自适应阈值处理和局部上下文池化的文档级关系抽取”。2020年。arXiv：2010.11304。[137] Z.-H. Zhou。“多示例学习：综述”。载于：南京工业大学计算机科学与技术系技术报告1（2004年）。

第 6 章
文本生成基础模型

摘要 本章讨论文本生成基础模型。这包括文档检索系统：这类系统接收查询，并从文档集合中返回按顺序排列的文本文件列表，通常通过评估嵌入的相似度来检索相关文本段落。问答系统接收自然语言问题，并须提供回答，通常以自然语言作答。机器翻译模型接收一种语言的文本，并将其翻译成另一种语言。文本摘要系统接收一篇较长的文档，并生成简短的总结，涵盖文档中最重要的内容。文本生成模型使用自回归语言模型生成较长的故事，通常以初始文本输入作为开头。对话系统的任务是与人类对话伙伴进行对话，通常不局限于特定主题。

关键词： 问答、机器翻译、文本摘要、文本生成、对话系统、文档检索

本章介绍基础模型，即用于在不同应用领域生成新文本的大型预训练语言模型。

- 文档检索系统接收一个查询，并从文档集合中返回按顺序排列的文本文件列表，通常通过评估嵌入的相似度来检索相关的文本段落（第 6.1 节）。
- 问答系统接收一个自然语言问题，并必须提供一个回答，通常以自然语言呈现（第 6.2 节）。
- 机器翻译接收一种语言的文本，并生成将其翻译成另一种语言的结果（第 6.3 节）。
- 文本摘要接收一份较长的文档，并撰写一份涵盖该文档最重要内容的简短总结（第 6.4 节）。
- 文本生成使用自回归语言模型生成较长的故事，通常从初始文本输入开始（第 6.5 节）。
- 对话系统的任务是与人类对话伙伴进行对话，通常不局限于特定主题（第 6.6 节）。

表格 6.1 通过示例说明的语言生成任务。

Task	描述	示例
文档检索	根据查询返回按顺序排列的文本列表文档。	新冠肺炎？ → wikipedia/covid-19,www.cdc.gov/,\protect.\kern\fontdimen3\font.\kern\fontdimen3\font.\kern\fontdimen3\font
生成式问答	生成问题的回答，通常使用一些背景知识。	阿尔伯特·爱因斯坦发明了什么？ → 爱因斯坦提出了相对论。
翻译	为源语言中的文本生成一个目标语言中含义相同的文本。	弗里茨喜欢吃火腿。 → 弗里茨喜欢吃火腿。
摘要生成	为长文本生成简洁的摘要。	那是隆冬时节，…… → 白雪公主被王子唤醒，后来她与王子结婚……
文本生成	从一段初始文本出发，生成连贯的续写文本。	贝多芬出生于波恩。 → 他的父亲是公爵的宫廷歌手。…
对话回答生成	在对话中生成一致的响应基于先前话语的序列。	你能推荐一部适合今晚观看的视频吗？ → *Netflix* 上有《记忆碎片》。

由于方法众多，我们将重点介绍撰写本文时性能出色的代表性模型。我们将评述各个领域当前最先进的技术，并根据适当的基准测试进行评估，同时考虑所需的计算资源。对于标准模型，我们提供了指向前文相关描述的链接。表格 6.1 展示了各应用领域的示例。

6.1 文档检索

信息检索 (信息检索) 使用计算机系统搜索数据库中的内容。由此产生的信息检索系统通常称为搜索引擎。通常，用户会针对某个主题拟定一个句子或查询，系统则需要返回与该查询相关的文档排序列表（即席检索）。这里我们聚焦于从存储的文档集合中检索文本信息。与第 6.2 节中的问答方法不同，该系统不会用自然语言直接生成对查询的回答。

早期的信息检索系统是基于关键词的：文档中包含的所有单词都会存储在倒排索引中。检索算法会搜索索引，以识别包含查询词的文档。随后，根据文档中每个查询词的信息含量对这些文档进行排序，例如使用 tf-idf 或 BM25 [186] 衡量。图 6.1 展示了这两个步骤。 [2] 对早期检索技术进行了综述。然而，这种方法存在三个主要问题：

- 许多物体、活动或事件都可以用不同的单词来表达，这些单词称为同义词，例如*“drink”* 和*“beverage”* ，或者*“buy”* 和*“purchase”*。包含替代表达的文档不会通过关键词检索返回。像*“hehas tons of stuff to throw away”* 和*“heneeds to get rid of a lot*

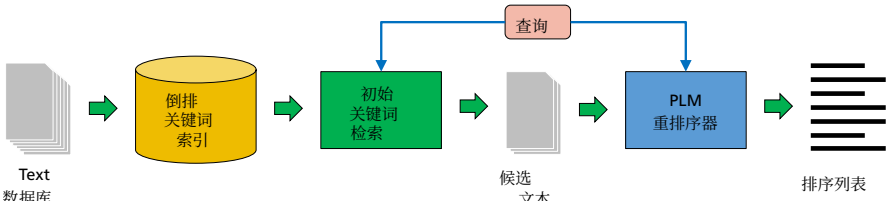


图 6.1 使用预训练语言模型的检索与重排序架构。首先，从文档集合中检索文本，通常使用精确匹配的词袋查询。然后，使用预训练语言模型嵌入（例如 BERT 生成的嵌入）对这些候选项重新排序。图片改编自 [123],，经作者慷慨许可转载。

of junk" 这样的改述更难被发现，因此也被忽略了。这被称为词汇不匹配问题。

- 许多词语会因上下文不同而有不同的意义（例如"rock"：摇滚乐或石头）。这类词语称为同形异义词。检索到的文档中，凡是包含这类词语的部分，都可能与查询不匹配。

- 词语顺序往往对句子的意义至关重要（例如"狗杀人"与"人杀狗"）。关键词搜索通常会忽略这一点。

作为替代方案，人们使用上下文嵌入来表示查询和文档。通过比较上下文语义表示来识别匹配的文档，可以减少文档与查询之间的词义差异，并检索包含同义词、同形异义词和释义的文本。这些模型在各种检索基准测试中取得了最先进水平的结果 [137]，最近也已应用于商业搜索引擎。因此，迄今为止，它们是预训练语言模型在商业上最重要的应用之一。

6.1.1 稠密检索

稠密检索方法会将文本编码为固定长度的嵌入向量，其长度远小于文本本身。文档是否与给定查询相关，取决于嵌入向量的相似度，该相似度通过余弦相似度或内积计算。与问答（第 6.2 节）不同，这些模型不会针对搜索查询生成直接的自然语言响应，而是返回完整文档或文本段落。最近，基于预训练语言模型的稠密检索方法在使用一小组域内相关性标注文档进行微调后，表现优于基于关键词的方法。Lin等人 [124] 全面概述了基于预训练语言模型的检索系统。稠密检索的不同方法可分为几类，将在后续章节中介绍：

- **交叉编码器**：将拼接后的查询和文档作为 BERT 的输入，并确定文档与查询的相关性（第 6.1.3 节）。

表 6.2 文档检索模型及其性能。基准测试（第 6.1.2 节）：MARCO：MS-MARCO [16], NQuest：Natural Questions 基准 [109], Wiki65K：较长的维基百科文档 [247]。

模型	描述	基准测试
monoBERT（第 6.1.3 节）	使用 BERT 处理每个查询-段落对。	MARCO 35.9% MRR@10
monoT5（第 6.1.3 节）	使用 T5 处理每个查询-段落对。	MARCO 38% MRR@10
ColBERT（第 6.1.4 节）	基于词元嵌入	MARCO 36.7% MRR@10
模型 1（第 6.1.4 节）	计算查询是文档“翻译”的概率的概率。	MARCO 39.1% MRR@100
SMITH（第 6.1.4 节）	使用基于 BERT 的分层编码器	Wiki65K 准确率 95.9%
SentenceBERT（第 6.1.5 节）	用于查询和文档的 BERT 编码器	将召回时间从 65 小时缩短至 5 秒
DPR（第 6.1.5 节）	用于查询和文档的不同 BERT 编码器，经过微调以降低检索损失。用于近似最近邻搜索的 FAISS 索引	NQuest 前 20 准确率为 79.4%
RocketQA（第 6.1.5 节）	用于查询和文档的 RoBERTa 编码器。之后重排序	MARCO 41.9% MRR@10
coCondenser（第 6.1.5 节）	使用 CLS 的查询和文档 RoBERTa 编码器词元。之后进行重排序	MARCO 40.8% MRR@100

- **使用词元嵌入进行检索**：查询和文档中的词元通过上下文嵌入进行编码。然后使用不同的指标比较这些嵌入，并收集相关文档（第 6.1.4 节）。
- **使用段落嵌入进行检索**：这些技术通过嵌入对查询和文档中的段落进行编码。随后，对这些嵌入进行比较。这种嵌入类型会考虑词序，因此有望返回更匹配的结果（第 6.1.5 节）。

这里只能介绍极少数方法，以便让读者了解当前使用的方法，如表 6.2 所示。第 6.2.2 节和第 6.2.3 节将讨论用于问答的检索技术，这些技术的能力更强。Lin等人 [124] 对用于检索的预训练语言模型进行了非常全面的综述。

6.1.2 文本检索性能测量

有许多用于训练和比较检索方法的基准数据集。*MS-MARCO*基准 [16] 是一个大规模集合，由从必应搜索查询日志中抽取的约 50 万个匿名化问题构成。对于段落排序任务，它包含一个由 880 万个段落组成的语料库，这些段落平均长度为 55 个词，提取自 360 万份网页文档。目标是检索能够回答问题的段落。训练集包含约 50 万对查询和相关文档，以及另外 4 亿对查询和不相关文档。此外，还有一个开发集和一个秘密测试集，其中各包含

约 7,000 个查询。不过，有讨论认为，MS-MARCO 基准的金标准标注在一定程度上存在偏见 [10]。

The *Natural Questions* （*NQ*） [109] 包含真实用户向 Google 搜索引擎提出的问题，每个问题至少有 8 个词。它要求问答系统阅读并理解整篇维基百科文章，而文章可能包含也可能不包含问题的答案。例如，问题是“血液离开右心室后会被泵送到哪里？”任务是检索一个长答案，即页面中回答该问题的一个段落，例如“血液从右心室经由肺动脉半月瓣泵出.....”，或者指出没有答案。该任务旨在贴近端到端问答应用。人工标注员提供一到五个答案。最初的 Natural Questions 基准是一项阅读理解任务，为每个问题提供若干证据文档；而 ***EfficientQA*** 基准 [147] 则通过选取最多包含五个词元答案的示例并舍弃证据文档，将其改编为开放域问答。

Min 等人 [146] 指出，Natural Questions 中超过一半的查询存在歧义，歧义来源多种多样，例如事件和实体引用。他们构建了一个*AmbigQA*，其中包含经过改写且能够得到唯一回答的问题。

一种简单的评估指标是*top-k* 准确率，即检索器至少返回一个正确回答的查询所占的比例。更复杂的指标是平均倒数排名 （*MRR*），即第一个正确回答排名的倒数；如果没有返回正确回答，则为 0。例如，如果第三个回答是正确的，则倒数排名为 1/3。对于 $|Q|$ 个查询，MRR 为

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}. \tag{6.1}$$

$MRR@m$ 表示始终返回一个有序的 m 文档列表。

我们可以将 $Pr(i)$ 定义为列表中前 i 个元素所达到的精确率，该列表大小为 m，即前 i 个元素中相关文档所占的比例。然后，我们可以将平均精确率定义为

$$AP = \frac{1}{m}\sum_{i=1}^{m} Pr(i) * rel(i) \qquad MAP = \frac{1}{|Q|}\sum_{j=1}^{|Q|} AP_j \tag{6.2}$$

其中，如果第 i 个文档相关，则 $rel(i) = 1$，否则为 0。平均精度均值（*MAP*）是对 $|Q|$ 个不同查询的 AP 求平均。

6.1.3 使用 BERT 的交叉编码器

monoBERT [155] 根据基于 [*CLS*] 词元嵌入进行微调的 BERT 分类器执行重排序。查询和文档组合为输入

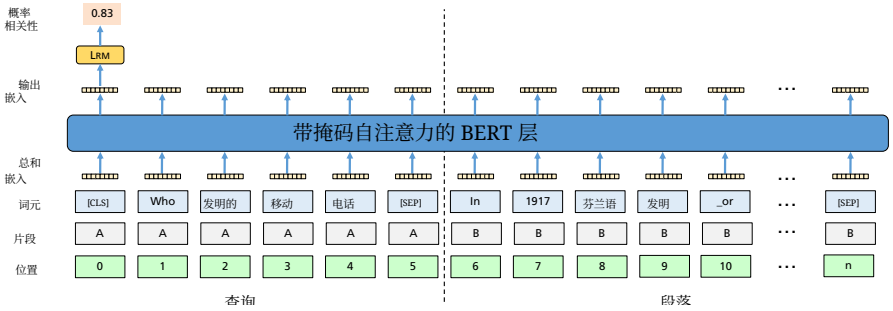


图 6.2 monoBERT模型使用经过微调的BERT模型，根据查询对段落进行排序。输入由查询与段落拼接而成。 [*CLS*] 词元嵌入经过训练后，能够返回该段落回答查询的概率。

“[*CLS*] <查询> [*SEP*] <文档> [*SEP*]”。该输入由在 MS-MARCO 上微调的 BERT 处理，最后一层中 [*CLS*] 的嵌入由逻辑分类器用于预测当前文档与查询相关的概率。该输出得分用于排序（图 6.2）。请注意，通过这种技术，诸如“流感的症状包括发烧和鼻塞”和“鼻塞和体温升高是你可能患流感的迹象”之类的释义可能会被识别出来。

在MS-MARCO 基准上， [153] monoBERT的MRR@10值达到35.9%（即第一个相关文档平均位于第2.8位）。相比之下，此前基于关键词的BM25搜索的MRR@10值为16.5%（第一个相关文档平均位于第6.1位），因此这一结果使搜索引擎的性能实现了大幅提升。无论在学术界还是工业界，由单个模型带来如此显著的效果跃升都十分罕见，这也使社区立即为之振奋。

monoBERT 为文本排序问题提供了一种简单而有效的解决方案，这一点相当令人瞩目（至少对于短于其最大输入长度的文本而言） [124]。多项研究发现，在词频保持不变时，monoBERT 估算相关性的表现优于 BM25。通过改变现有文档的文本操纵测试发现，调整句子内部或句子之间的词语顺序会产生显著的负面影响，而打乱文档内句子的顺序则会产生较小的负面影响。相比之下，仅调整前置词几乎没有影响。输入模板变体的实验结果表明，monoBERT 会利用精确匹配、“软”语义匹配以及词语的位置信息。对于不同类型的查询、不同语料库以及不同设置等情形，这些不同组件究竟如何结合，仍是一个悬而未决的问题。需要注意的是，这种搜索方法需要消耗大量计算资源，因为每个段落都必须重新进行评估，而索引搜索所需的工作量仅按对数增长。

monoT5 [154] 使用 T5 编码器-解码器模型而非 BERT，对检索到的文档重新排序。模型接收的输入为 “查询： <查询> 文档：<文档> 相关：”。monoT5 经过微调后，会在文档与查询相关时生成词元 *true* ，不相关时生成 *false*。预测为 *true*的概率可用作相关性分数。对于拥有 30 亿个参数的 T5，作者在 MS-MARCO 段落检索任务上取得了 38% 的 MRR@10 值。这表明，更大的模型能够提升检索系统的性能。

6.1.4 使用词元嵌入进行检索

每个 Transformer 编码器层中的 BERT 注意力模式具有全对全的性质，这意味着相对于输入长度而言，时间和空间复杂度均呈二次增长。在第 3.2 节中，我们介绍了若干应对较长输入的方法。这些方法都可以用于处理较长文档。在众多方法中，我们将更详细地讨论 ColBERT 和模型 1。

ColBERT [99] 对另一个（成本更低的）检索模型的输出进行重新排序，该模型通常是基于术语的模型；或者直接对文档集合执行端到端检索。查询和文档前面分别添加了不同的特殊词元。ColBERT 使用单个预训练BERT模型，将每个查询或文档编码为词元嵌入袋。在最后一层中，嵌入的维度会被缩减，并归一化为欧几里得长度 1.0。因此，内积等价于余弦相似度。如果（q_1、……、 q_m）是查询词元，而 $d_{i,1}$、……、$d_{i,k}$ 是第 i 个文档的词元，则 q 与 d_i 的相似度计算为

$$s_{q,d_i} = \sum_{r=1}^{m} \max_j \boldsymbol{\eta}(q_r)^\intercal \boldsymbol{\eta}(d_{i,j}). \tag{6.3}$$

这是查询中的每个词项与文档 d_i 中包含的“最佳”匹配词项之间的最大余弦相似度（MaxSim）之和。对于每个查询嵌入，都会考虑 L2 距离最近的 10 个嵌入，并检索 $k = 1,000$ 个最近的文档向量。

为了对初步搜索结果（例如 1,000 篇文档）进行排序，需要计算所有查询嵌入与检索到的文档中的所有嵌入之间的最大相似度（例如余弦相似度）。这种方法非常高效，因为与之前的方法相比，它所需的每秒浮点运算次数少了几个数量级。在 MS-MARCO 基准 [153] 上，采用重排序的 ColBERT 可达到 34.9% 的 MRR@10 值（第一个相关文档平均位于第 2.9 位），略低于交叉编码器 monoBERT。

ColBERT 也可用于端到端检索。它在准备阶段使用 *FAISS* 索引 [91] ，存储文档词元嵌入，以便进行 k 最近邻搜索。请注意，每个文档中的每个词元都必须存储一个嵌入，因为嵌入取决于上下文。检索分为两个阶段：第一阶段，对每个查询词元执行多次近似搜索

。第二个细化阶段会根据 MaxSim 准则对这些近似匹配结果重新排序。在 MS-MARCO 基准上，ColBERT 的端到端检索 MRR@10 值为 36.7%，远优于重排序性能，与成本高得多的 BERT 交叉编码器方法不相上下。

模型 1 [28] 结合了多种基于词元嵌入的检索模型技术。首先，作者估计查询$\boldsymbol{q}$ 作为文档 $\boldsymbol{d}$的“翻译”生成的概率 $p(\boldsymbol{q}|\boldsymbol{d})$。利用贝叶斯定理，作者得到

$$p(\boldsymbol{d}|\boldsymbol{q}) \propto p(\boldsymbol{q}|\boldsymbol{d})p(\boldsymbol{d}) \propto p(\boldsymbol{q}|\boldsymbol{d}) \tag{6.4}$$

假设均匀先验 $p(\boldsymbol{d})$ [21]。他们考虑查询词元 q_i是文档词元 d_j的翻译的概率 $r(q_i|d_j)$。通过神经网络近似 $r(q_i|d_j)$，他们使用词元 q_i 和 d_j 的嵌入作为输入，并能够估计 $p(\boldsymbol{d}|\boldsymbol{q})$。该方法所需的计算开销很小。作者将 BERT 稠密检索器与 Lucene 搜索索引结合起来。最后，他们使用 Doc2query 扩展Model 1的文档。*Doc2query* [156] 旨在生成查询，使文档与这些查询相关。该方法训练 Transformer，从最多包含 400 个词元的文档中生成最多 100 个查询词元。模型使用由查询和相关文档构成的数据集进行训练，例如 MS-MARCO。在 MS-MARCO 上，他们取得了 39.1% 的 MRR@100。无上下文的神经模型 1 不如基于 BERT 的排序模型有效，但可以在 CPU 上高效运行（无需昂贵的索引时预计算，也无需在查询时对大型张量执行运算）。

目前，还没有检索器尝试处理长文档。这有许多重要应用，例如新闻推荐、相关文章推荐和论文引文推荐。通常，长文档会被划分为段落，其基本思想是相关内容包含在某个段落中。请注意，使用更长输入的预训练语言模型，例如 BigBird，可以提升性能（第 3.2 节）。不过，显然仍需对这一点进行评估。**SMITH**模型 [247] 使用基于 BERT 的分层编码器来捕获文档结构信息。文档首先被划分为句子，并计算每个句子的词元嵌入。每个句子都以一个 [*CLS*]词元开头，其嵌入表示该句子。更高层的句子级 BERT 只接收句子嵌入作为输入。第二层 BERT 的第一个人工词元被用作整个文档的嵌入。

模型通过掩码语言建模任务进行预训练，以获得词元嵌入。此外，在第二层中，还有一个掩码句子块预测任务，模型必须从一个批次中的所有句子嵌入中选出正确的嵌入。对于经过相关性标注的微调数据集，微调任务通过逻辑分类器根据文档嵌入预测相关性得分，并使其最大化。在包含长篇维基百科文章的 *Wiki65K* 上 [87] ，该方法的准确率达到 95.9%，显著优于以往的方法。

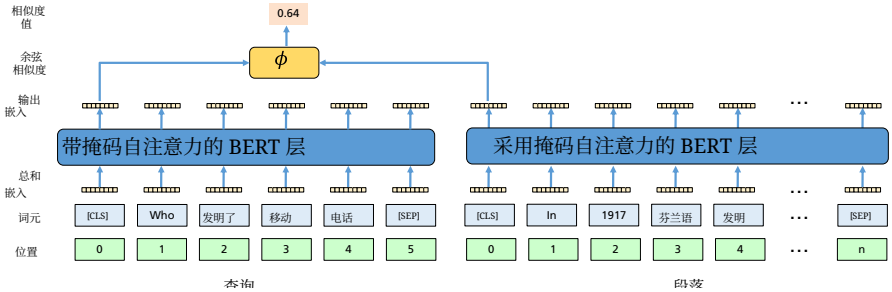


图 6.3 SentenceBERT 模型使用两个微调后的 BERT 模型，将查询和段落转换为 [*CLS*] 词元的嵌入。随后，使用余弦相似度模块计算相似度值。

6.1.5 稠密段落嵌入与最近邻搜索

通过嵌入向量表示文本段落，有望在表示空间中直接匹配“意义”，从而解决词汇不匹配问题。这些所谓的稠密检索 技术可直接对预训练语言模型生成的向量表示进行排序。该方法有望解决词汇不匹配问题。与计算成对差异相比，这种方法提供了效率高得多的检索过程。具体而言，通过索引和近似最近邻搜索，将查询的嵌入向量与段落的嵌入向量进行匹配。如今，开源库中已有高效且可扩展的解决方案。

给定一个查询 q 和一组文档 $D = \{d_1, \ldots\ldots, d_n\}$，我们希望定义函数 $\boldsymbol{\eta}_q(\cdot)$ 和 $\boldsymbol{\eta}_d(\cdot)$，将词元序列 q 和 d 转换为固定宽度的向量。这些函数应具备以下性质：如果 d_i 与查询 q 相关，则 $\boldsymbol{\eta}_q(q)$ 与 $\boldsymbol{\eta}_d(d_i)$ 之间的相似度最大。我们希望估计

$$p(\text{relevant} = 1 | d_i, q) := \phi(\boldsymbol{\eta}_q(q), \boldsymbol{\eta}_d(d_i)), \tag{6.5}$$

其中，$\phi(\cdot)$ 是相似度比较函数，例如标量积 [124, p。 133]。请注意，$\boldsymbol{\eta}_d(d_i)$ 可以预先计算并组织到索引中。通过为查询和文档使用不同的编码器 $\boldsymbol{\eta}_q(\cdot)$ 和 $\boldsymbol{\eta}_d(\cdot)$，我们可以考虑查询和文档不同的角色与措辞。

SentenceBERT [183] 是用于生成具有语义意义的句子嵌入的双编码器设计原型，可用于大规模文本相似度比较（图 6.3）。查询 q 和文档 d_i 由同一个 PLM（BERT 或 RoBERTa）进行处理。相似度通过余弦相似度进行比较

$$\phi(\boldsymbol{\eta}_q(q), \boldsymbol{\eta}_d(d_i)) = \frac{\boldsymbol{\eta}_q(q)^\intercal \boldsymbol{\eta}_d(d_i)}{\left\|\boldsymbol{\eta}_q(q)\right\| * \|\boldsymbol{\eta}_d(d_i)\|}. \tag{6.6}$$

为生成句子嵌入，作者研究了三种方案。（1）使用 [*CLS*] 词元的嵌入。（2）对所有输出嵌入进行平均（平均池化）。（3）对所有输出嵌入逐分量取最大值（最大池化）。未进行微调时，结果比非上下文嵌入更差。微调提升了性能，并达到了新的最先进水平。结果表明，平均池化是最有效的设计，略优于最大池化或使用[*CLS*] 词元。最重要的是，在 10,000 份文档中查找最佳匹配的计算时间从 65 小时缩短到了 5 秒。

DPR [94]为查询 q和约含 100 个词的文本段落 d_i 分别使用了独立的编码器 $\boldsymbol{\eta}_q(q)$和 $\boldsymbol{\eta}_d(d_i)$。两个编码器都将 $\text{BERT}_{\text{BASE}}$ 的 [*CLS*] 嵌入作为其输出表示。$_{\intercal}$作为比较函数，使用了内积$\boldsymbol{\eta}_q(q)\ \ \boldsymbol{\eta}_d(d_i)$。对于每个查询 q_i，训练集包含一个正确的段落d_i^+和若干负段落$d_{i,1}^-$，……，$d_{i,m}^-$。损失函数编码了以下目标：对于 q_i 和d_i^+取得较大的 ϕ值（即相似度），而对于 q_i 和d_i^-取得较小的相似度，$_j$

$$L(w) = -\log \frac{\exp[\boldsymbol{\eta}_q(q)^\intercal \boldsymbol{\eta}_d(d_i^+)]}{\exp[\boldsymbol{\eta}_q(q)^\intercal \boldsymbol{\eta}_d(d_i)] + \sum_{j=1}^{m} \exp[\boldsymbol{\eta}_q(q)^\intercal \boldsymbol{\eta}_d(d_{i,j}^-)]} \tag{6.7}$$

负例由以下几类段落混合构成：通过关键词搜索检索到但不包含回答的段落，因此属于困难负例；此外，还使用了同一训练批次中其他示例的段落。我们可以采用近似最近邻搜索，而不必穷举计算$\eta_q(q)$与$\eta_d(d_i)$之间所有文档的相似度。*FAISS* [91] 是一种基于分层可导航小世界图的开源方法。在 Natural Questions 基准上，他们取得了 79.4% 的前 20 准确率，显著优于基于关键词的 BM25 搜索此前 59.1% 的前 20 准确率。复现研究 [136] 证实了这些结果，但发现 DPR 与 BM25 的混合方法可以将性能提升至 82.6%。

ANCE [238]使用单个 RoBERTa 模型对查询和文档进行编码。训练期间，通过在由训练所得编码器生成的表示所构建的索引上进行近似最近邻搜索，选取困难负例。这样便能选出“难以区分”的负例。索引会定期更新。在 Natural Questions 上，ANCE 的 top-20 准确率达到 82.1%。研究还将其性能与 monoBERT 交叉编码器进行了比较：后者使用 monoBERT 对第一阶段 BM25 的结果重新排序，方法是逐一比较所有文档与查询。结果显示，在 MS-MARCO 上，将 monoBERT 应用于 BM25 的 MRR@10 为 34.7%，而 ANCE 为 33%。显然，交叉编码器的效果优于 ANCE。作者还使用大小为 64 的嵌入和近似最近邻搜索，将 ANCE 应用于 80 亿篇文档。据报告，与此前的商业实现相比，性能提升了 16%。

RocketQA [184] 首先执行检索步骤，随后进行重新排序。两种方法通过列表式训练方法联合优化，即使用由正例和负例组成的列表训练这两个模块。此外，他们还通过结合随机采样和去噪采样进行数据增强，以构造多样化的训练实

例。在 MS-MARCO 段落检索任务中，他们报告的 MRR@10 为 38.8%。随后对排名前 50 的结果进行重排序，可将 MRR@10 提高至 41.9%。

coCondenser [63]是 MS-MARCO 排行榜 [140]中排名最高的条目之一。该模型被迫学习将信息聚合到 *"CLS"*嵌入中，随后该嵌入将参与语言模型预测。接着使用额外的"对比损失"：来自同一文档的段落的*"CLS"* 嵌入应彼此相似，而来自不同文档的段落嵌入之间的距离应更大。这样便能为段落生成表现力很强的嵌入。当模型在 MS-MARCO 上进行微调后，其在 MS-MARCO 排行榜 [140]上的 MRR@100 回报为 40.8%。

可用实现

- DPR 代码可在以下地址获取：https://github.com/facebookresearch/DPR。
- FAISS 最近邻搜索的代码可在 https://github.com/facebookresearch/faiss 获取。
- 使用 ANCE 代码和数据训练的最近邻搜索可在 https://github.com/microsoft/ANCE 获取。
- RocketQA 的代码和数据可在此处获取：https://github.com/PaddlePaddle/RocketQA。
- FlexNeuART [27]实现了 Model 1 检索系统 [28]。
- coCondenser 的代码位于：https://github.com/luyug/Condenser。

6.1.6 总结

检索是网络搜索中的关键步骤，其任务是从包含数十亿篇文本的语料库中找出一小组与查询相关的候选段落。在检索阶段发现更多语义相关的候选项，有望向最终用户呈现更多高质量结果。稠密检索方法代表了搜索引擎技术的一次范式转变。它们能够识别词语和释义的意义，从而找到与查询匹配得好得多的段落。搜索结果还可用于问答模型（第 6.2 节）和对话系统（第 6.6 节）。必应 [35, 238, 266], 、Google [152, 197],和 Facebook [82]已经在生产环境中的搜索引擎里使用了这些方法。

上述讨论的稠密检索方法在监督式设置下进行微调，并使用人工相关性标签作为输入，例如来自 MS-MARCO 的标签。使用两个不同的预训练语言模型分别对查询和文档进行编码，可以获得最佳结果。训练这两个预训练语言模型的目标，是提高正确参考文档相对于某些负样本文档的概率。由于使用两个不同的预训练语言模型需要付出更多努力，大多数系统使用单个模型对问题和文档进行编码。实验表明，稠密检索与关键词检索的结合似乎具有优势。

第 6.2.2 节和第 6.2.3 节讨论了用于问答的检索技术，这些技术甚至更加强大。

一个问题是搜索系统向新领域的可迁移性。当以类似于 monoBERT [124, p. 72] 的方式使用时，研究发现 BERT 具有很强的跨域相关性分类能力。如果使用来自一个领域（例如推文）的相关性判断对 BERT模型进行微调，那么它可以成功应用于不同的领域（例如新闻电讯稿）。另一方面，Thakur 等人 [221]创建了一个名为 *BEIR* 的基准测试，其中包含来自生物医学和推文等截然不同领域的 18 个检索任务。作者在 MS-MARCO 上训练了大量稠密检索技术，然后在其他任务上对其进行评估。他们发现，这些技术的效果平均低于 BM25；由于 BM25 简单易用，在大多数情况下都能发挥作用。

嵌入索引的内存需求不容忽视。对于包含 880 万个段落的 MS-MARCO 段落语料库，关键词 Lucene 索引需要 661 MB，而大小为 768 的向量所对应的 FAISS 索引需要 42 GB，ColBERT 索引则需要 156 GB [124, p. 159]。要将这些技术应用于网络规模的数据，需要采用内存占用更小的方法。

6.2 问答

问答 (QA) 是自然语言处理的一种应用，它接收自然语言查询，并自动以自然语言生成精准回答。这是一项可以追溯到 20 世纪 60 年代的长期人工智能任务 [69]。与第 6.1 节讨论的搜索引擎相比，问答系统会直接呈现问题的最终回答，而不是返回相关摘要片段或超链接列表。因此，它更加便于用户使用且效率更高。通常，系统可以访问文档数据库或知识库(*KB*)，例如维基百科，并在其中搜索相关信息。

封闭域问答系统 处理特定领域的问题，例如医学问题，并拥有关于该领域的背景知识，或使用涵盖该领域的大型训练集进行训练。开放域问答系统 (ODQA) 处理几乎任何主题的问题，通常依赖通用知识库或互联网搜索 [37]。多模态问答系统处理不同媒体中的问题，例如文本和图像。Zhu 等人 [265] 对 ODQA 进行了综述。表 6.3 汇总了领先的问答模型及其性能。

问答的一种简单形式是阅读理解，系统需要在给定文本中找出问题的答案。通常，类 BERT 系统会通过片段预测（第 2.1.3 节）在文本中标出答案片段。这项任务基本上可以视为已解决。对于*SQuAD 2.0* 基准测试 [179]，ALBERT 的 F1值超过 93%；经过微调的*ST-MoE-32B* 混合专家模型（第 3.5.2 节）拥有 269B 个参数 [270]，F1值达到 96.3%，而人工的 F1值为 89.5% [178]。不过，Sen 等人 [199] 指出，在一个数据集上训练的系统可能无法很好地泛化到其他基准测试。

表 6.3 问答模型及其性能。下半部分包含检索模型。基准测试：NQ：Google 查询的 Natural Questions 基准 [109], TriviaQA：TriviaQA 基准[92, 226],HotpotQA：多跳基准测试 [249],EM：精确匹配。

模型	详细信息	基准测试
BigBird（第 6.2.1 节）	自编码器，支持长输入，使用问答进行监督训练对	带参考文档的自然问题 57.9% 精确匹配 WikiHop 82.3% 准确率
PoolingFormer（第 6.2.1 节）	采用两级注意力架构的监督式自编码器使用问答对进行训练	自然问题使用参考文档，精确匹配率为 61.6%
RealFormer（第 6.2.1 节）	带旁路注意力的自编码器，使用监督训练问答对、多跳问答	WikiHop 84.4% 准确率
GPT-3（第 6.2.1 节）	大型自编码器 175B，仅预训练	自然问题 少样本 29.9% TriviaQA 少样本 71.2%
Gopher（第 6.2.1 节）	大型自编码器280B，仅预训练	自然问题少样本 28.2%
PaLM（第 6.2.1 节）	大型自编码器 540B，仅预训练	自然问题少样本 36.0% TriviaQA 少样本 81.4%
DPR（第 3.4.5 节）	采用两个 BERT 模型和 FAISS 索引的检索器-阅读器	NQ 精确匹配准确率 41.5% TriviaQA 57.9%
FiD（第 3.4.5 节）	采用 T5 模型和 FAISS 索引的检索器-阅读器	NQ 精确匹配准确率 51.4% TriviaQA 67.6%
REALM（第 3.4.5 节）	基于 BERT嵌入点积的检索器-阅读器，速度较慢，NQ精确匹配准确率为 40.4%	
FB HYBRID（第 3.4.5 节）	DPR检索器与其他检索器结合，FiD阅读器	NQ精确匹配 ACC 为 53.9%，相当于 67.4% 的正确率
MS UNITED（第 3.4.5 节）	基于 BERT 的检索器，T5+ELECTRA 作为阅读器，最终重新排序	NQ精确匹配准确率为 54.0%，对应于 65.8% 的正确率
AISO（第 3.4.5 节）	采用重复检索轮次的检索器-阅读器，多跳问答 HotpotQA72.0% F1值	
RETRO（第 6.2.3 节）	语言模型，使用冻结的BERT检索器，语言模型定期包含检索到的词元块	NQ精确匹配 acc 45.5%
WEBGPT（第 6.2.3 节）	与必应搜索引擎结合的 GPT-3，它可以被定期调用	TriviaQA 69.5%

6.2.1 基于训练数据知识的问答

语言模型通常基于综合文本集合进行训练，因此能够记住大量信息。一个常用的基准测试是自然问题（*NQ*） [109], ，其样本取自谷歌搜索日志（第 6.1.2 节）。对于给定的问题，系统需要在给定的支持文档中找到一个简短答案片段。例如，问题“啤酒花何时加入酿造过程？”的答案应为“煮沸过程”。

*TriviaQA*基准测试 [92, 226] 包含一组最初从 Web 抓取的问答题。与 Natural Questions 不同，这里的问题是围绕已知答案编写的。*TruthfulQA* [125] 是一种特殊的问答基准测试，包含以对抗方式构造的简短问题，因此有些人的回答可能会因错误信念和误解而出错。答案根据信息丰富度和真实性进行评估。

微调问答模型

BigBird （第 3.2 节）自注意力被用作自编码器，并以掩码语言模型目标进行训练，输入序列包含 4,096 个词元 [253]。在 Natural Questions 上微调时，模型必须在给定的某篇证据文档中找到一个简短答案片段。模型在此任务上取得了 57.9% 的 F1值。**PoolingFormer** [256] 是一种适用于长输入序列的替代模型，采用两级注意力架构。第一级使用较小的滑动窗口模式，从邻居处聚合信息。第二级通过池化注意力使用更大的窗口来扩大感受野。由微调后的 PoolingFormer 组成的集成在 Natural Questions 基准上取得了 61.6% 的 F1值。该模型与 **SMITH** 模型相似 [247], ，后者使用基于 BERT 的分层编码器来捕捉文档结构信息（第 6.1.4 节）。

另一种选择是**Macaw** [218], ，这是一个可免费使用的问答系统，拥有 11B 个参数。它基于 T5 构建，具备强大的零样本问答能力。在一组包含 300 道挑战性问题的测试中，作者声称 Macaw 的表现比 GPT-3 高出 10%，尽管它的参数量只有 GPT-3 的一小部分。除了回答问题，Macaw 还可以根据答案生成问题，或为答案和问题生成多项选择选项。作者还提供了详细的错误分析。

要整合不同的证据来找到答案，难度要大得多。用于测试这种能力的一项基准测试是*WikiHop* [232], ，它要求合并来自不同文档的信息。一个示例问题是“孟买空中花园位于哪个国家？”，相关文档包括“孟买的空中花园也称为 *Pherozeshah Mehta Gardens*，是梯田式花园……”以及“孟买是印度马哈拉施特拉邦的首府，也是印度人口最多的城市……”。对于每个查询，都会向模型提供最多 140 个背景段落。在这项基准测试中，BigBird-ETC（第 3.2.1 节）的准确率达到 82.3%。目前，这项任务的最佳模型是 **RealFormer** ，准确率为 84.4%[171], ，略低于人类 85% 的表现。RealFormer 是一种采用改进架构的自编码器，它将前一层所有注意力头的原始注意力分数旁路传递到后续层 [76]。

使用少样本语言模型进行问答

近期的基础模型使用海量文档集合进行训练，无需输入额外知识便能生成问题的答案。一个例子是自回归语言模型**GPT-3** ，它拥有1750亿参数，曾在约5000亿词元的书籍、维基百科和网页文本集合上进行预训练（第3.1.2节）。由于模型容量很大，它能够将大量“知识”吸收到参数中。当基础模型不允许使用外部信息时，这称为闭卷问答。

提示：
我是一个高度智能的问答机器人。如果你问我一个有事实依据的问题，
我会给出回答。如果你问我一个无意义、欺骗性的问题，
或者没有明确答案的问题，我会回答“未知”。
问：美国的人均预期寿命是多少？
答：美国人的预期寿命为78年。
问：1955年谁是美国总统？
答：德怀特·D·艾森豪威尔于1955年担任美国总统。
问：他属于哪个政党？
答：他属于共和党。
问：香蕉的平方根是多少？
答：未知
...
问：帝王谷在哪里？
A:

回答：
帝王谷位于埃及卢克索。

图 6.4 一种可能的少样本提示，可让 GPT-3 根据预训练期间获得的现有知识给出回答 [160]。

如（第3.6.3节）所述，可以通过少量示例（少样本）指示GPT-3完成一项任务。图6.4给出了一个少样本提示示例。对于Natural Questions，GPT-3在零样本设置、单样本设置和少样本设置下的精确匹配准确率分别为14.6%、23.0%和29.9% [29, p. 14]。这一结果是在未针对Natural Questions进行微调的情况下取得的。拥有2800亿参数的更大型**Gopher** 模型（第3.1.2节）表现略差，在少样本设置下的准确率为28.2% [175,p. 80]。

规模更大的 **PaLM** 模型拥有 5400 亿个参数（第 3.1.2 节），其训练使用了包含 7800 亿个词元的高质量数据集。该模型采用一种新的提示技术来提出逻辑问题：将示例与思维链 一同呈现给系统，把推理任务划分为更小的问题（第 3.6.4 节）。通过这种方式，它获得了将来自不同来源的事实结合起来、得出最终回答的方法。

PaLM 还在大量其他基准测试上进行了评估，其中一部分属于问答任务。在 Natural Questions 上，它在零样本条件下的准确率达到 21.2%，在少样本提示下达到 36.0% [43,p。 47]。在 Trivia QA（关于维基百科的问题）、BoolQ（使用是/否答案的问答）和 PIQA（需要推理的问答）上，PaLM 也达到了新的最先进水平。结果见表 3.4。PaLM 接受了大量测试，其中包括 150 多项 BIG-bench 任务（第 4.1.4 节）。其中许多是与问答相关的任务：21 项上下文问答任务、24 项无上下文问答任务、36 项阅读理解任务，以及大量涉及特定知识和常识的任务 [1, 22]。PaLM 在问答基准测试上的其他结果见 [43, p。 12], ，其中 PaLM 始终达到最先进水平。

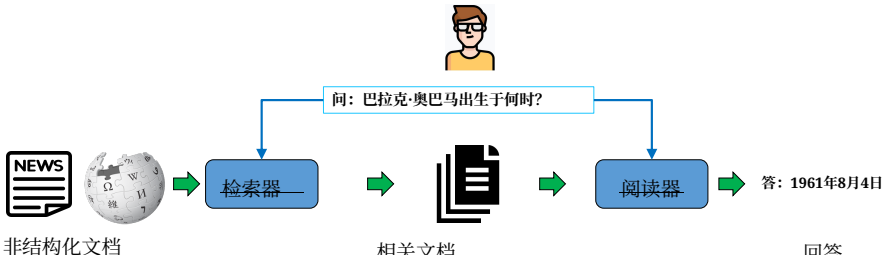


图 6.5 问答通常结合稠密检索与答案选择模块。检索器通过比较查询的嵌入与段落的嵌入来执行稠密检索。阅读器对检索到的文档进行排序，并通过自回归预训练语言模型 [36]生成回答。表 A.2 中列出了图像各部分的出处。

6.2.2 基于检索的问答

检索式开放域问答系统通常分两个阶段运行：对于一个问题，检索器模块会从文本集合中找到若干可能包含答案的文档。随后，阅读器 会考虑该问题和检索到的文档，并生成自然语言答案（图 6.5）。由于该模型依赖外部信息，因此被称为开卷问答。

第 3.4.5 节介绍了检索器，并在第 6.1 节的文档检索背景下进行了讨论。检索器可以采用使用词频-逆文档频率加权或 BM25 的传统搜索引擎。或者，检索器也可以是基于文档和问题嵌入的稠密检索器。它通过计算嵌入相似度来检索段落，例如使用*DPR* [94]（第 3.4.5 节）。 [36]提供了关于开放域问答的教程。

阅读器 通常是自回归语言模型，它将查询和检索到的文档同时作为输入。它经过微调，可以根据检索信息及其内部知识生成针对查询的响应。

使用外部知识库进行问答的优势在于，经过整理的知识库通常会经过正确性检查。不过，它们对实体的覆盖范围可能有限，关系也可能不是最新的。目前有多种方法使用实体映射等技术（第 3.4.1 节）将预训练语言模型与知识库结合起来。近期论文提出了一种结合知识库与检索的方法 [239]。知识引导的文本检索 [145] 首先为查询检索文本段落。它会创建一个段落图，其中顶点是文本段落，边表示从外部知识库或同一文章中的共现关系中推导出的关系。在 Natural Questions [109] 上，该方法达到了 34.5% 的准确率。

HYBRIDER [41] 既利用检索器提供的信息，也利用结构化知识库和表格中的信息。作者收集了维基百科页面，并构建了名为 HybridQA 的基准数据集，其中包含需要借助文本、表格和超链接进行多跳推理的问答对（图 6.6）。模型首先将问题链接到表格单元格，以及维基百科文本段落和超链接。

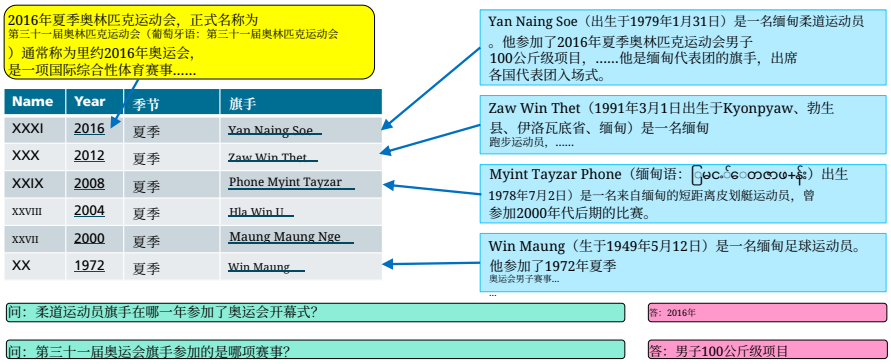


图 6.6 对于混合式问答，HYBRIDER 会检索维基百科页面 [41]（左上）。部分页面包含表格（左侧）。这里可以解读列标题，也可以将其视为指向实体的超链接（带下划线）。下方显示了两组人工标注的问答对。经作者 [41, 许可转载图像。第 2] 页。

在推理阶段，系统会对已链接的信息进行排序和整合，从而得出不同答案的概率。使用该数据集进行的实验表明，仅利用表格或仅进行检索，精确匹配准确率约为 20%；而联合模型的准确率则超过 40%。不过，混合模型的得分仍远低于人类表现。

最早的检索器—阅读器系统之一是 **DPR**（密集段落检索器）[94]。它使用 BERT 模型将段落编码为嵌入，并借助 FAISS 索引通过近似 k 最近邻搜索检索段落（第 6.1.5 节）。这样便能找到意义相近但措辞不同的段落。DPR 阅读器是另一个 BERT 模型，经过微调后，可为每个检索到的段落预测一个概率，表示该段落包含正确答案的可能性。此外，它还通过片段预测选出一段词元，这段内容可能就是答案。此方法可轻松应用于包含数十亿段落的知识库 [94, 213]。在 *NaturalQuestions* [109] 上，其测试集准确率为 41.5%。

FiD [84] 在第 3.4.5 节中进行了介绍。检索器基于 DPR，并比较查询与段落的嵌入。Raffel 等人 [177] 已表明，T5 等生成模型能够为问答任务生成回答。FiD 通过基于 aT5 模型的 阅读器 处理查询和检索到的段落，以生成回答。由于第一步是逐个处理段落，因此该系统非常高效。与 DPR 的 41.5% 相比，FiD 在 Natural Questions 测试集上取得了 51.4% 的精确匹配准确率。

REALM [75] 和 **RAG** [114] 是用于开放域问答的检索增强生成模型。然而，它们会在自回归语言模型中同时处理所有检索到的段落，无法考虑大量段落，因此在 Natural Questions 上的准确率较低：

REALM 为 40.4，RAG 为 44.5。Sachan 等人 [194] 提出了一种用于检索增强开放域问答的端到端可微训练方法。潜变量指示应纳入哪些相关文档。潜变量的值通过期望最大化算法迭代估计。在 Natural Questions 上，他们取得了 52.5% 的精确匹配准确率。

MTR [138]基于这样的观察：神经检索器在其微调领域表现良好，但通常在域外性能方面表现不佳。作者提出了一种类似于 DPR 的多任务检索器，并在八种不同的检索任务上进行联合微调。他们使用共享的段落编码器，因此只需一个编码后段落的索引即可，同时还使用一个在所有任务之间共享的查询编码器。在八个模型中的五个模型上，其性能超过了针对相应领域进行微调的专用模型。

AISO [268] 是一种用于解决多跳问答任务的检索器-阅读器架构，这类任务需要多个文档才能回答问题。系统会进行重复检索轮次，并将相关术语作为新的搜索查询，以寻找更多证据。该方法具有自适应性，并且在每一步都会选择三种检索操作之一（例如 BM25、DPR 和超链接），或选择一种答案操作。在*HotpotQA*基准 [249], 中，问答系统必须在整个维基百科范围内找到查询的答案。AISO 模型以 72.0% 的联合 F1值创下新的最先进水平。

FB HYBRID 系统在 EfficientQA 竞赛中亮相 [147],，该竞赛旨在解决 Natural Questions 数据集中的真实用户谷歌搜索引擎查询问题 [109] 。原始的 NQ 是一项阅读理解任务，会为每个问题提供若干证据文档；而 EfficientQA 基准测试 [147] 将其改为开放域问答：只保留答案不超过五个词元的示例，并舍弃证据文档。该系统采用检索器-阅读器架构 [158]。检索器融合了 DPR 和另一种检索系统，后者能够处理列表、表格和知识库关系，并检索 100 个段落。阅读器是一个 T5-large 序列到序列模型，它接收检索器提供的 100 个段落并生成答案。背景语料库包含来自维基百科的 1880 万个段落。在 Natural Questions 上，该模型的精确匹配准确率达到 53.9%。根据人工评估者进行的评估，该模型能够正确回答 67.4% 的问题，表现与使用搜索引擎的人类专家大致相当。MS UnitedQA 模型采用了类似的架构 [139]。它使用基于 BERT 的检索器，以及由 T5 模型和 ELECTRA 组合而成的阅读器；阅读器会处理返回的文档并生成不同的答案。最后由重排序模型选出答案。MS UnitedQA 在 Natural Questions 上的精确匹配准确率为 54.0%，正确率为 65.8%。如果将系统的内存占用限制为 6 GB，其性能仅略有下降。

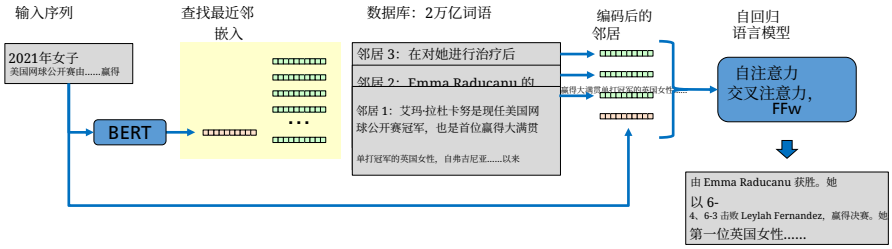


图 6.7 Retro 语言模型为输入序列检索信息。该模型将输入序列和从数据库中检索到的邻居块作为输入，并生成适当的回答 [176]。

6.2.3 使用检索进行长文本问答

具有集成检索的语言模型

Retro [25] 是一个拥有 70 亿参数的自回归语言模型，它利用检索信息来预测下一个词元。它使用冻结的 BERT 模型作为检索器（图 6.7）。每个训练序列都被拆分为多个块，并通过从包含 2 万亿个词元的数据库中检索出的最近邻 k 对这些块进行扩充。返回的信息会由语言模型处理，以改进下一个词元的预测，从而大幅提升性能。阅读器由一个可微自回归编码器和一个分块交叉注意力模块组成，用于预测词元。

长度为 $n=2,048$ 个词元的输入序列 $\boldsymbol{v} = (v_1, \ldots\ldots, v_n)$ 被分割成长度为 $m=64$ 的块 $\boldsymbol{c}_t = (v_{(t-1)*m+1}, \ldots\ldots, v_{t*m})$。每个块 $\boldsymbol{c}_t$ 都通过从数据库中检索到的一组 $\text{Ret}(\boldsymbol{c}_t)$ 个 k 最近邻块进行扩展。随后，下一个块中的词元 v_{t*m+i} 的概率 $\boldsymbol{c}_{t+1}$ 可以递归计算为

$$p(v_{t*m+i}|v_{t*m+(i-1)}, \ldots, v_{t*m+1}, \boldsymbol{c}_t, \text{Ret}(\boldsymbol{c}_t), \ldots, \boldsymbol{c}_1, \text{Ret}(\boldsymbol{c}_1)). \tag{6.8}$$

$(t+1)$ 个块的第 i 个词元的概率 $\boldsymbol{c}_{t+1}$ 仅取决于先前的词元，以及针对先前块从数据库中检索到的数据 $\text{Ret}(\boldsymbol{c}_j)$。这将检索过程集成到了语言模型中。

块 $\boldsymbol{c}_t$ 的检索器使用 $\boldsymbol{c}_t$ 中所有词元的 BERT嵌入的平均值 $\text{Bert}(\boldsymbol{c}_t)$ 作为键。它根据 L_2 距离 $||\text{Bert}(\boldsymbol{c}_t) - \text{Bert}(\tilde{\boldsymbol{c}}_s)||_2^2$，从数据库中检索 k 个最近邻。模型接收相应的块 $\tilde{\boldsymbol{c}}_{s,\,j}$ 以及它们后续的续写块 $\tilde{\boldsymbol{c}}_{s+1,\,j}$，共计 $j=1、\ldots\ldots、k$ 个，这些块共同构成 $\text{Ret}(\boldsymbol{c}_t)$ 的元素。通过过滤，可以避免将待预测的块包含在 $\text{Ret}(\boldsymbol{c}_t)$ 中，否则将使条件概率的定义失效。检索使用 *SCaNN* 库 [73], ，以 $O(\log T)$ 时间完成；该库能够在 10 毫秒内从包含 2 万亿个词元的数据库中收集一组块。请注意，Retro 的文档语料库约比 FiD 和其他检索模型的数据库大 1,000 倍。

在阅读器内部，Ret(c_t) 中检索到的词元被输入自编码器，自编码器计算出一组 E 编码后的邻居。随后，所谓的 Retro 模块

$$\textsc{Retro}(H, E) := \textsc{Fcl}(\textsc{Catl}(\textsc{Attl}(H), E)), \tag{6.9}$$

以及标准的自注意力模块 Lm(H) := Fcl(Attl(H)) 交错排列，并作用于中间嵌入 $H \in \mathbb{R}^{n \times d}$。其中，Fcl(·) 是一个全连接层，Attl(·) 是一个自注意力层，而 Catl(·， E) 是一个交叉注意力层，其中包含 E 中的信息。这些模块的输入和输出维度为$\mathbb{R}^{n \times d}$。

由此得到的语言模型能够高度可靠地预测下一个词元。*Pile* 数据集 [62] 是一个 825GB 的开源文本集合，由 22 个多样且高质量的数据集组成。该数据集经过有害语言和偏见筛查，例如针对性别、宗教和种族的偏见。其作者建议使用每字节位数（*bpb*）来衡量词元预测的质量；与困惑度不同，每字节位数不依赖于分词器 [62, p。 6]。作者将 Retro 与 GPT-3_{175B}[29], Jurassic-1_{178B} [121] 以及 Gopher$_{280B}$ [176]进行了比较。结果表明，在 5 个 Pile 数据集上，Jurassic-1 的 bpb 值最低（表现最佳）；在 2 个数据集上，Gopher 最佳；在 9 个数据集上，Retro 最佳，尽管它的规模远小于其他模型 [25]。GPT-3 不如这三个模型中的任何一个。这些结果的一个潜在问题是，检索语料库与测试数据存在重叠。

对于 *LAMBADA* 基准测试 [165]，模型必须预测一个段落的最后一个单词。作者测量了以下准确率：无检索的 Retro 为 70%，带检索的 Retro 为 73%，Gopher 为 74.5%，GPT-3 为 76.2%。请注意，Retro 的参数量仅为 GPT-3 的 4%。对于问答而言，Natural Question 基准测试具有相关性。在该测试中，Retro 达到了 45.5% 的精确匹配准确率。

LaMDA [222]对话系统（第 6.6.3 节）是 Retro 的扩展版本，拥有 137B 个参数。它展示了检索模型能够提升事实性。此外，它还能够通过一套过滤器屏蔽不需要的语音，从而减少有害语言。尽管该模型也可以轻松用于问答，但尚无相应的基准测试结果可知。

通过预训练语言模型控制搜索引擎

WebGPT [149] 扩展了 GPT-3，使其能够控制 必应搜索引擎 并针对特定查询执行网络搜索。语言模型必须发出诸如“搜索…… ”、“在页面中查找：…… ”或“引用：…… ”之类的指令，如图 6.8 所示。通过这种方式，模型从包含与问题相关信息的网页中收集段落。使用必应的优势在于，它具备强大的搜索能力，并且能够覆盖大量最新文档。

浏览会持续进行，直到模型发出结束浏览的命令、参考文献总长度达到最大值，或动作数量达到最大值。如果检索到相关参考文献，模型就会生成一段较长的回答来回答问题。

命令	效果
搜索 <查询>	将 <查询> 发送至必应 API，并显示搜索结果页
点击了链接 <链接 ID>	通过给定 ID 的链接前往新页面
在页面中查找：<文本>	查找 <文本> 的下一个匹配项并滚动到该处
引用：<文本>	如果在当前页面中找到 <文本>，则将其添加为参考文献
向下滚动了 <1，2，3>	向下滚动若干次
向上滚动了 <1，2，3>	向上滚动若干次
Top	滚动到页面顶部
Back	转到上一页
结束：回答	结束浏览并进入回答阶段
结束：<无意义内容，有争议>	结束浏览并跳过回答阶段

图 6.8 WebGPT 语言模型可能采取的动作。如果生成了其他文本，则该动作无效，会被忽略 [149]。

首先对 GPT-3 模型进行微调，使其模仿人工演示，从而能够使用基于文本的浏览器回答问题。随后，通过微调奖励模型来预测人类偏好，并通过拒绝采样对其进行优化，以提升模型答案的实用性和准确率。具体而言，模型经过微调，用于回答来自子版块“Explain Like I’m Five”的开放式问题数据集中的问题。*ELI5* [56], 示例见图 6.9。所提出的 WebGPT 答案应当连贯、相关，并有可信文档作为支撑。文中没有报告 GPT-3 的输入提示中是否包含当前搜索状态，也没有说明 GPT-3 模型如何将返回的文档整合为答案。不过需要注意的是，ELI5 的训练集与验证集之间存在显著重叠：至少 81% 的 ELI5 验证问题以经过限定的形式出现在训练集 [106] 中。

最终答案是通过排序，从 1750 亿参数 WebGPT 模型的 64 次试验中选出的。人工评估者在 69% 的情况下更偏好这些答案，而非 ELI5 数据集中的参考回答。此外，在 56% 的情况下，他们更偏好这些答案，而非人工示范者回答。

对于 WebGPT，针对 TruthfulQA [125] 的响应约有 75% 的时间是诚实的，而 GPT-3 在使用有帮助的提示时得分为 64%。GPT-3 的答案约有 20% 的时间既诚实又信息丰富，而 WebGPT 的最佳版本将这一比例提高到了约 56%。由于人们对 94% 的问题都给出了诚实的回答，因此模型的性能仍存在显著差距。在 TriviaQA 上，WebGPT 的得分为 69.5%，远低于 PaLM 的 81.4%。

一项创新的特征是通过参考文献为文本段落提供支撑。这与科学论文通过参考文献支撑主张的方法相对应， [143] 也曾提出过这一点。参考文献解释了回答，并支撑陈述的事实准确性。引文由必应根据查询选取。因此，它们应当与最终的读者生成的回答密切相关，并提供一种便捷的方式来评估响应的正确性。

然而，作者指出，参考文献并不总能代表可用证据，尽管模型引用的参考文献与生成文本相对应。此外，模型难以验证参考文献的可信度。在此，可以采用 Web-of-Trust 系统和搜索引擎技术，优先选择经过信任核查且被频繁链接的网页。

问题：
隐形眼镜如何保持在虹膜上？是什么阻止它们在
整只眼睛上到处移动？

回答：
隐形眼镜通过附着在眼睛的泪膜上来保持在适当位置，并且
通过眼睑的压力[1]。隐形眼镜
之所以能固定在眼睛里，是因为它们会附着在
表面的泪液层上[2]。泪液始终浮在眼睛表面，
而眼睑的压力有助于让隐形眼镜“粘附在泪液上”
并起到润滑作用[2]。眨眼时，你会为眼睛提供润滑，
并冲走可能会粘在镜片上的杂质
嗯[2]。你的隐形眼镜可能会因多种原因脱落，
包括佩戴不合适以及过于用力地揉眼睛[3]。

[1]。什么是你的隐形眼镜？Optometrics.org
[2]。佩戴非处方隐形眼镜是否更安全？Campbell、Cunningham、Taylor 和 Haun 医生
[3]。为什么我的隐形眼镜总是从眼睛里掉出来？www.self.com

图 6.9 WebGPT 生成的长篇问题回答。系统自动从 64 个回答中选出最佳答案。引文由必应搜索引擎自动检索，并添加到回答中 [80]。

可用实现

- BigBird 代码和模型可在 https://huggingface.co/google/bigbird-roberta-base
- DPR 代码和模型 https://github.com/facebookresearch/DPR
- FiD 代码和模型 https://github.com/facebookresearch/FiD
- RealFormer 代码 https://github.com/jaketae/realformer
- REALM 代码 https://github.com/google-research/language/blob/master/language/realm/README.md
- RETRO 的 PyTorch 实现，DeepMind 的检索式注意力网络。本实现与论文略有不同，采用旋转位置嵌入进行相对位置编码，并使用 FAISS 库替代 SCaNN https://github.com/lucidrains/RETRO-pytorch。

6.2.4 总结

目前已有多种基础模型问世，能够提升问答性能。例如，自回归语言模型 GPT-3（175B）、Gopher（175B）和 PaLM（540B）拥有庞大的参数集，这些模型在大型文档集合上训练，能够获取丰富的知识。借助少样本提示，它们无需使用外部知识即可高准确率地回答问题。

近年来，检索器-阅读器架构越来越多地应用于问答系统。它有望利用更大的知识库或互联网，而且这些内容可以轻松保持最新。检索器可以采用关键词搜索或稠密检索。稠密检索能够缓解术语不匹配问题，避免忽略相关释义。通常会预先计算每篇文档或每个短语的嵌入，并事先构建嵌入索引。当前系统借助 FAISS 和 SCaNN 等先进的近邻搜索引擎比较嵌入，从而访问多达万亿级词元的文档集合。

阅读器通常以文本形式接收查询和检索到的段落，并生成答案。它经过微调，能够选出正确答案，并提供表达清晰、真实可信的答案。Retro模型是一种仅有70亿个参数的自回归语言模型，它使用冻结的BERT模型检索到的段落作为额外的当前状态信息来生成下一个词元。它能够在许多问答任务中大幅提高准确率，也可用于故事生成等其他应用。

WebGPT将GPT-3与必应搜索引擎相结合，以检索文档并生成恰当的答案。它能够通过引用文档来增强生成文本，这些文档可为答案提供依据并加以说明。LaMDA对话模型是Retro的扩展版本，拥有1370亿个参数，并经过专门调整以提高可用性和事实准确性。此外，它还能够通过一套过滤器屏蔽不需要的言论，从而减少有害语言。这些技术也可应用于问答。

答案的生成仍然很困难，因为正确的响应需要来自多个文档的信息。在这种情况下，必须进行多轮查询。RealFormer、HYBRIDER 或 AISO 等特殊模型可以提升 WikiHop 等基准测试的性能。

6.3 神经机器翻译

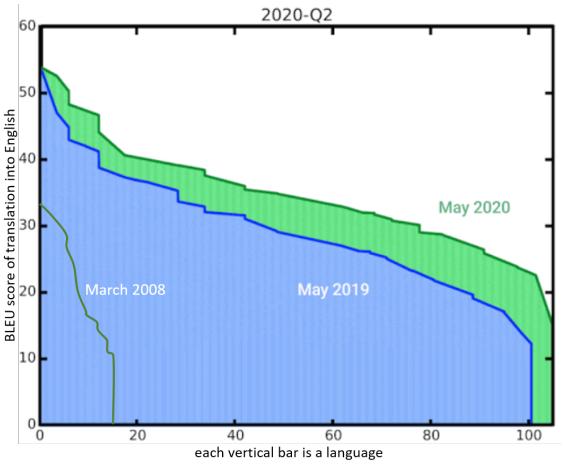
语言是大多数人类交流与互动的基石。此外，许多人以语言的方式进行思考，并利用语言表达和交流感受、目标与想法。我们通过语言交流知识，并利用语言建立社交和情感关系。世界上有超过 7,100 种语言 [19], 其中一些语言如图 6.10 所示。跨越语言障碍相互理解的能力，对于在人与人之间交流思想至关重要。

循环神经网络 [15, 215]初步取得成功后，Transformer 编码器-解码器（第 2.3 节）的发展推动了神经机器翻译（NMT）的进步。通过交叉注意力，可以建立源文本中每个词元与译文之间的“相关性”，从而生成比以往更好的译文。大型训练集的可用性以及更优的模型架构，持续提升了用于 NMT 的预训练语言模型的性能（图 6.11）。第 3.3 节介绍了多语言处理的标准模型。[248]提供了一篇综述。



图 6.10 这张地图展示了世界上 7,100 种语言，每个点代表一种语言，颜色表示该语言所属的主要语系。目前，世界上只有一小部分语言被基础模型涵盖。经作者 [24, p. 23] 慷慨授权转载。

图 6.11 Google 将 100+不同语言翻译成英语的 BLEU 分数随年份的变化。图像来源见表 A.2。



6.3.1 单一语言对翻译

NMT 的训练数据由源语言文本及其目标语言译文组成。传统上，评估是通过将一个或多个参考译文与待评估译文进行比较来完成的，正如综述 [195] 中所述。现有多种自动指标，例如 BLEU、METEOR 或 BERT 得分（第 2.3.3 节）。事实证明，人工判断与自动评估之间存在明显差异。因此，如今大多数高端比较都会请人工译者评估翻译方法的质量。

得分	捷克语	德语	豪萨语	冰岛语	日语	俄语	中文
到英语							
最佳模型 z 分数	FB **0.111**	BL**0.126**	FB 0.248	FB 0.293	HW 0.141	NV 0.137	NI 0.042
人工 z 分数	-0.085	-0.081				0.089	0.019
最佳模型 BLEU	43.1	53.0	18.8	40.6	27.8	56.3	33.4
来自英语							
最佳模型 z 分数	FB 0.263	OB**0.266**	FB 0.264	FB 0.594	FB 0.430	OW 0.277	HN 0.284
人工 z 分数	**0.397**	0.030	0.362	**0.872**	0.314	0.317	0.325
最佳模型 BLEU	33.6	31.3	20.4	30.6	46.9	45.0	49.2

表 6.4 WMT2021 新闻翻译任务的领先系统。系统按标准化 z 分数排序 [5, p. 15-19]。最佳系统或人工参考译文可能显著优于其他系统（以粗体标示）。系统：FB：Facebook-AI，BL：Borderline，HW：HW-TSC，NV：Nvidia- NeMo，NI：NiuTrans，OB：Online-B，OW：Online-W，HN：HappyNewYear。

在 WMT2021 机器翻译会议上，众多团队完成了将英文新闻文本翻译成德语、日语、俄语、中文以及若干低资源语言，或从这些语言翻译成英文的基准测试 [5]。与其使用 BLEU 等比较统计指标，每个系统的译文都由多位人工评估者进行评估，且评估者看不到参考译文。他们需要根据给定译文对相应源语言输入所表达意义的贴切程度，在模拟量表上进行评分；该量表对应的底层绝对评分量表范围为 0–100。由于有些评分者可能较为严格，因此系统按 z 分数排名，该分数以评分者为单位进行均值中心化和归一化。根据哪些系统显著优于其他所有系统，系统使用 Wilcoxon 秩和检验进行分组。为评估人工评估的有效性，研究投入了大量精力。

共收到 173 个提交系统。此外，还纳入了五个匿名在线系统。人工制作的其他参考译文在所有表格中均以“HUMAN”标注。结果表明，几乎所有优秀系统都基于 Transformer 编码器-解码器。词语主要由 SentencePiece [107] 分词器（第 1.2 节）进行编码。一种广泛使用的技术是 回译 [200]。在此过程中，先将单语文本翻译成另一种语言，然后再回译。通过最小化译文与原始文本之间的差异，可以改进两个模型。每种语言最多有 5 亿个句子可用于回译，这显著提升了质量。此外，在大多数情况下，集成模型能够提高性能。

表 6.4 展示了每种语言对的最佳系统结果。通常，排名靠前的模型有 2 至 5 个，其性能差异并不显著。Facebook-AI 模型（FB）在一半的语言对中取得了最佳结果。此外，还展示了根据 n 元语法自动计算出的最佳系统的 BLEU 分数。可以看出，对于形态丰富的语言（如捷克语和德语），总体而言，“译入英语”的值优于“从英语翻译”，尤其如此。与人工参考译文相比，最佳系统在三种语言对上的表现显著更好。这一点已经被批判性地讨论过

，[223], 对此提出批评，认为句子之间的上下文有限，而且评估人员的翻译能力有限。

通过增加参数数量提升了性能。例如，Facebook 模型 [224], 采用了一个包含 4.7B 个参数的标准模型，以及一个最多包含 128 个专家的稀疏门控混合专家系统。在每个稀疏门控混合专家层中，系统根据学习得到的门控函数的得分，将每个词元路由到得分最高的两个专家前馈块。此外，这些模型还使用来自新闻领域的领域特定数据进行了微调。使用束搜索生成 n 个最佳假设，再根据概率的加权平均值对其进行排序：p(tgt|src)、p(src|tgt) 和 p(tgt)，其中 src 是源句子，tgt 是目标句子。

众所周知，单句翻译存在歧义（例如代词或同形异义词），而结合文档上下文即可消除这些歧义。在 WMT2021 中，通过在文档上下文 [5] 中评估翻译质量，将这一点纳入了考量。当前的编码器-解码器基础模型能够处理更大上下文，因此有望提升翻译性能 [141]。我们不应寻找某个句子最可能的译文，而应为给定的完整源文档生成最佳译文。尽管比较句子级翻译通常无法体现人工翻译与机器翻译之间的差异，但比较文档级翻译往往会在统计上显著地偏向人工翻译 [110]。

与其使用输入序列特别长的序列到序列模型，**HAT** [187]提出了一种层次注意力Transformer。作者将输入文本拆分为句子，并在每个句子 i 前加上一个特定的 $[BOS_i]$ 词元。这些词元概括句子的内容，并通过常规的自注意力和交叉注意力与其他句子建立联系。对于 WMT2019 上从英语到德语的文档翻译，常规编码器-解码器Transformer的 BLEU 得分为 32.5，而 HAT 的 BLEU 得分达到了最先进水平：34.5。

6.3.2 多语言翻译

通常，训练数据稀缺的语言，其翻译准确率会低得多；表 6.4 中的豪萨语便是如此。近期的一项成功，是通过增强神经机器翻译的多语言能力来扩展其功能，这一点已在第 3.3 节讨论过。这显著提升了资源匮乏语言的翻译效果。综述请参见 [48]。

M2M 是 Facebook AI 推出的模型，通过大规模挖掘构建的监督数据，利用覆盖 100 种语言和数千个翻译方向、包含 75 亿个句子的庞大语料库，提升了多种语言之间的翻译效果。该模型采用 Transformer 编码器-解码器架构，拥有 150 亿个参数。作者在编码器中添加了一个特殊词元，用于指示源语言；在解码器中添加了一个特殊词元，用于指示目标语言。Transformer 包含 12 层编码器和 12 层解码器，嵌入维度为 1024。由于所有语言共用一个词元词汇表，因此输入和输出嵌入也共用。为提升性能，作者为五种语言在解码器中添加了语言专属层。

模型	捷克语	德语	豪萨语	冰岛语	日语	俄语	中文	
从英语								
FB-Mult	36.1	31.3	20.1	33.3	46.8	46.0	49.9	
WMT2021 最佳	33.6	31.3	20.4	30.6	46.9	45.0	49.2	
差异	**2.5**	0.0	-0.3	**2.7**	-0.1	**1.0**	**0.7**	
译为英语								
FB-Mult	43.5	53.3	21.0	41.7	27.7	57.1	32.1	
WMT2021 最佳	43.1	53.0	18.8	40.6	27.8	56.3	33.4	
差异	**0.4**	**0.3**	**2.1**	**1.1**	-0.1	**0.8**	-1.3	

表 6.5 Facebook 多语言模型与提交至 WMT2021 新闻任务的最佳语言对模型的 BLEU 分数。所报告的数值为最终 WMT2021 测试集上的 BLEU 分数 [225]。如果多语言模型表现更好，则以粗体标出模型之间的差异。

借助特定的并行化技术，他们仅使用数百个 GPU 就训练出了该模型。

除四个语言方向（英语→中文、中文→英语、英语→芬兰语、英语→爱沙尼亚语）外，该模型在 WMT 基准测试上的翻译结果平均提升了 1.9 个 BLEU 得分。区域语言的提升尤为显著，平均增加了 7.6 个 BLEU 得分。对于资源丰富的语言对，Liu 等人 [130]建议使用极深的 Transformer，最多包含 60 个编码器层和 12 个解码器层。他们开发了一种简单而有效的初始化技术，可稳定训练，并在 WMT2014 英语-法语任务上以 46.4 BLEU 的成绩达到最先进水平。

尽管多语言翻译具有许多优势，但对于高资源语言对而言，其性能通常不如专门训练的双语模型。最近，Facebook [225] 提出了一个单一的多语言模型，在WMT2021 新闻基准的14个语言对中，有10个语言对的性能超过了最佳的专门训练双语模型。Facebook构建了两个多语言系统：任意语言到英语，以及英语到任意语言。他们采用数据挖掘技术，从大规模网页爬取数据中识别翻译数据，并利用来自全部八种语言的可用单语数据，其中包含数亿个句子，以最大限度地提升机器翻译系统的性能。他们对可用的单语数据进行过滤，以减少噪声数量，然后使用现有的最强多语言模型集成对这些数据进行回译。参数数量从150亿增加到530亿，以增强模型容量。

BLEU分数显示在表6.5中。与WMT2021中最佳的双语模型相比，多语言模型在14个案例中的9个案例中取得了更高的BLEU分数，这表明来自其他语言的额外训练数据有助于翻译。只有中文→英语的BLEU分数下降幅度较大，下降了1.3分。作者还对英语→俄语和英语→德语这两个语言对进行了人工评估。结果表明，在双语翻译和多语言翻译的质量之间，受试者并未感知到差异。

表 6.6 展示了所采用的改进策略对多语言模型中不同语言的影响。对于豪萨语和冰岛语这类训练数据很少的语言，回译的效果显著。然而，作者指出，回译——

改进策略	捷克语	德语	豪萨语	冰岛语	日语	俄语	中文			
双语	33.1	38.7	14.7	25.8	25.4	25.8	40.0			
+ 回译	33.1	39.6	23.1	29.4	26.1	25.7	42.4			
+ 微调	35.7	39.5	23.3	29.4	27.7	26.0	43.0			
+ 多语言	36.4	40.8	24.6	31.2	29.7	26.8	43.6			
+ 集成	36.8	41.1	25.0	32.5	29.7	26.9	43.6			
+ 重排序	37.2	41.1	25.5	32.8	29.7	27.4	43.6			
+ 后处理	39.8	42.6	25.5	34.5	29.8	28.8	48.2			

表 6.6 Facebook AI 提交的 WMT2021 方案 [225] 在 WMT2021 开发集上的 BLEU 分数受到不同建模改进的影响。最后一行所示的版本已提交。

表 6.7 PaLM 少样本翻译性能与专用模型先前微调后的翻译性能及先前少样本性能的比较。

左侧展示的是传统 WMT 语言对中从英语译出和译入英语的翻译结果。右侧展示的是英语与哈萨克语（kk）之间的双向翻译，以及德语与法语之间的翻译 [43, 第 27] 页。

from	en	en	en	fr	de	ro	en	de	kk	fr
to	fr	de	ro	en	en	en	kk	fr	en	de
先前微调后的最先进水平	45.6	41.2	33.4	45.4	41.2	39.1	15.5	31.5	30.5	24.9
先前少样本最先进水平	33.9	26.8	20.5	38.8	40.6	37.3	-	-	-	-
PaLM 540B 少样本	**44.0**	**37.4**	**28.7**	**42.8**	**47.5**	**43.8**	5.1	25.7	20.8	17.4

翻译会通过在一种语言中生成罕见的人工短语，产生翻译腔 。在特定领域（例如新闻文本）上进行微调，或可减轻这些影响。这样做可使译成英语的 BLEU 得分提高约 3 分，译出英语的 BLEU 得分提高 0.7 分。切换到多语言模型可提升所有模型的表现。虽然模型集成的效果不大，但使用条件目标源概率对束搜索翻译结果重新排序，可使 BLEU 得分提高约 0.4 分。后处理（例如应用标准标点规则）可能产生显著影响，例如中文可提高 5 个 BLEU 得分。

拥有 540B 参数的 **PaLM** 自回归语言模型，其 780B 个训练词元中约有 22% 来自非英语训练文本 [43]（第 3.1.2 节）。与其他大型语言模型类似，PaLM 并未专门使用平行文本进行训练，不过这类数据可能自然存在于训练语料库中。表 6.7 将 PaLM 540B 的少样本翻译结果与先前的少样本结果及微调后的最先进水平 [43, p. 27]进行了比较。每个语言对中 BLEU 值最高的结果以带下划线标示，少样本 BLEU 值最高的结果则以粗体显示。表中显示，在传统的 WMT 翻译对上，PaLM 始终取得最佳少样本 BLEU，且往往以明显优势胜出。对于低资源语言哈萨克语（kk），微调后的翻译模型的 BLEU 值优于 PaLM。不过，在 de→en 和 ro→en 这两个语言对上，PaLM 的表现优于监督模型。此外，PaLM 对 fr→en 的零样本翻译取得了 25.2 的 BLEU 值，高于微调后的最先进水平 24.9。总体而言，尽管 PaLM 并未针对这项任务进行训练，其表现仍接近微调模型。

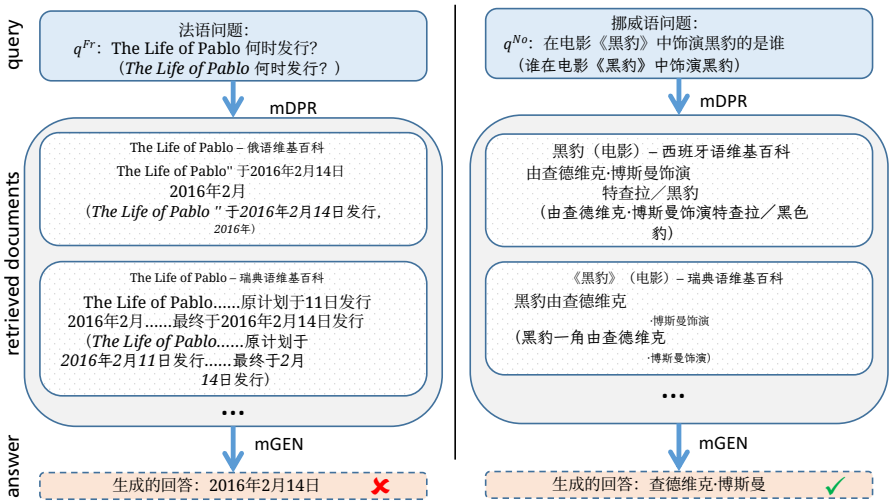


图 6.12 CORA系统使用 mDPR 进行跨语言检索，并使用 mGEN 生成答案 [13, 第 9] 页。问题的答案是正确的，但左侧的答案本应使用法语作答。

6.3.3 多语言问答

近年来，开放域问答（ODQA）发展迅速（第 6.2 节）。因此，将这些技术扩展到多语言问答极具价值。这样一来，便能利用世界上约 6,900 种语言中编码的信息，并通过为使用较少见语言的人们提供答案来缩小数字鸿沟。Ruder [192, 193]编写了一篇关于多语言 ODQA 的教程。

进行多语言开放域问答的一种简单方法是将问题翻译成英语，使用英语开放域问答系统生成答案，再将答案翻译回目标语言。由于翻译存在歧义，这一流程在某些情况下可能会产生错误 [132]。此外，与目标语言相关的信息以及目标文化中的概念化方式可能无法在英语中体现 [258]。

TyDiQA-GoldP 基准是一种问答数据集，涵盖 11 种类型学上各不相同的语言，包含 204K 个问答对。涵盖的语言包括：英语、阿拉伯语、孟加拉语、芬兰语、印度尼西亚语、日语、斯瓦希里语、韩语、俄语、泰卢固语和泰语。由于该基准测试所涵盖的语言具有非常多样的结构，因此可以预期，在这些数据上表现良好的模型在其他语言上也能取得较高的问答准确率。MKQA 是一个评估数据集，通过将 10k 个 Natural Questions 翻译成 25 种目标语言而创建。

作为另一种方案，可以训练跨语言检索器和阅读器模型，结合多种语言的信息，以目标语言生成答案（图 6.12）。CORA 能够跨多种语言回答问题，即使某些语言没有专门的标注数据或知识来源也不例外。它包含一个密集段落检索器，可为问题检索不同语言的文档。使用 mBERT（第 3.3.1 节）的预训练多语言语言模型 mDPR 经过微调，分别对段落和问题进行编码。通过执行最大内积搜索，检索出最相似的前 k 个文档，与 DPR（第 3.4.5 节）类似。研究表明，mBERT 能提升非英语单语检索的搜索质量。阅读器 mGEN 是一种多语言自回归序列模型（例如 mT5，第 3.3.2 节），它汇总检索到的段落中的信息，以目标语言生成答案。该方法不使用专门的翻译模型。初始训练数据由多语言问答数据集组合而成。这些数据集中的每个训练样本都包含一个问题、一个正面段落和一个答案。然而，这些数据集在语言多样性方面存在局限。因此，作者利用不同语言维基百科文章之间的链接，迭代生成更具代表性的低资源语言训练数据。

结果表明，在涵盖 26 种语言的多语言开放域问答基准测试中，CORA 的表现大幅优于此前的最先进水平，其中有 9 种语言在训练期间未曾出现。CORA 可将平均 F1值从 17.1 提高到 21.8。使用 mDPR 进行检索时，在使用拉丁字母文字的印欧语言中表现良好，即使这些语言未曾出现于训练数据中也是如此。对于使用非拉丁字母文字的语言（如日语、俄语和中文），性能则大幅下降。或许在这些情况下，模型无法利用其他语言中的相关段落来回答问题。

mT5 （第 3.3.2 节）是 T5 Seq2seq Transformer 的多语言版本，拥有多达 13B 个参数 [246]。它使用涵盖 101 种语言、约含 48B 个词元且共用 25 万个词元词汇表的网页训练数据集进行预训练。在 TyDiQA 基准上微调后，其精确匹配得分达到 79.1%。**ByT5** [245] 是 mT5 多语言编码器-解码器的一种变体，拥有 12.9B 个参数。它处理的是 UTF-8 字节，使用包含 256 种可能字节值的词汇表，而非词元。该模型经过预训练，可替换平均长度为 20 字节的受损跨度。最大的模型使用 36 个编码器层和 12 个解码器层。在所有目标语言的黄金数据上微调后，该模型在 TyDiQA 基准上的精确匹配得分达到 81.4%。

PaLM 基础模型 [43] 的 7800 亿个训练词元中，约有 22% 来自非英语训练文本（第 3.1.2 节）。因此，它可用于翻译和问答等多语言任务。使用少样本提示时，它在 TyDiQA 上的精确匹配得分为 60.5%。在 TyDiQA 上对该模型进行微调后，得分提高到 80.0%，略低于 ByT5 XXL 的性能。表 6.8 中的详细结果展示了该模型在不同语言上的表现。PaLM 在其中两种语言上的得分高于 ByT5。作者指出，与 PaLM 相比，ByT5 使用的非英语文本多出 50%，这或许可以解释这一差异。

模型	阿拉伯语	孟加拉语	英语	芬兰语	印度尼西亚语	韩语	俄语	斯瓦希里语	泰卢固语	平均值
mT5 XXL	76.9	80.5	75.5	76.3	81.8	75.7	76.8	84.4	83.9	79.1
ByT5XXL	**80.0**	**85.0**	**77.7**	78.8	**85.7**	**78.3**	**78.2**	84.0	**85.5**	**81.4**
PaLM 540B 微调	75.0	83.2	75.5	**78.9**	84.1	75.7	77.1	**85.2**	84.9	80.0
PaLM 540B 少样本	56.4	54.0	65.5	66.4	69.2	63.8	46.8	75.6	46.9	60.5

表 6.8 在包含 11 种类型逻辑不同语言的 TyDiQA 问答基准测试上与最先进水平的比较。数值为验证集上的精确匹配准确率 [43, p. 32]。

可用实现

- Hugging Face 提供 Marian、BART 和 T5（最多 110 亿个参数），以及多语言版 mBART 和 mT5 的实现与训练模型 https://huggingface.co/transformers/。
- M2M-100 [55] 提供开源数据收集脚本、模型代码以及训练模型的参数。此外，Fairseq 系统 https://github.com/pytorch/fairseq 可免费使用。
- CORA [13] 的多语言问答实现、生成的训练数据和训练模型可在此获取：https://github.com/AkariAsai/CORA。

6.3.4 总结

近年来，机器翻译取得了飞速发展。采用编码器-解码器预训练语言模型能够克服循环神经网络架构的局限，并将性能提升至接近人工水平。除了使用编码器-解码器 Transformer，网络爬虫借助基础模型获取高质量训练样本，以及采用特定评估程序，也是取得进展的原因 [33]。句子回译带来了进一步改进，尤其提升了低资源语言的结果；此外，训练一个用于所有语言之间翻译的单一多语言模型，也带来了改进。使用适当的并行化策略，训练参数量高达 600B 的多语言翻译模型，可显著提升 100 种语言的性能，BLEU 评估结果亦证实了这一点 [113]。近期，多语言模型的表现甚至超过了高资源双语翻译模型。PaLM基础模型也体现了这一点：在某些语言对上，其少样本翻译性能优于先前微调的模型。因此，多语言模型很可能会在未来成为标准。不过，当前使用无监督多语言训练的多语言模型，可能无法充分深入地刻画各种语言及语言变体的细微差异。未来的应用中需要对此加以检验。

这些发展为多语言问答系统带来了机会，例如 CORA，用户可以使用大量语言提出查询。

答案汇编自多种语言中的可用信息。通过这种方式，可以将并非所有语言都具备的文化特征和概念纳入考量。这也与跨语言语义解析密切相关：在跨语言语义解析中，自然语言话语会被转换为逻辑形式，以便在某个知识库中执行并返回答案 [202]。同样，PaLM基础模型为多语言问题提供的少样本答案，在相同基准测试上的准确率可与微调模型相媲美。PaLM 的一个微调版本甚至能够超越两种语言上最先进水平的微调模型。

然而，机器翻译问题尚未解决。训练数据与测试数据之间仍存在领域不匹配的问题。在某些情况下，系统无法准确捕捉句子的意义。如果不同语言对性别的处理方式不同，系统可能会生成有偏文本。另一方面，注意力机制使解码器能够直接查看远处的文本，并免费提供词语之间的软对齐。最近，通过翻译整篇文档，模型性能得以提升，因为仅凭句子往往不足以消歧所有单词。要将当前的多语言模型扩展到数千种语言，还需要新的技术 [19]。一种方法是使用单语数据集来改进翻译，因为可用单语文本的数量比译文多出几个数量级。此外，这还需要高度可靠的语言检测器，它们也能用于低资源语言。

6.4 文本摘要

表格 6.9 摘要模型的性能，以 ROUGE-2 衡量。基准测试包括：CNN/DM：CNN/Daily Mail 基准 [78]；XSum [151]，将新闻文章概括为一个句子；arXiv [46]，长篇科学文档；PubMed [46]，长篇医学文档；Multi-News [54]，平均文档长度为 1,793 个词，每个簇包含 2.8 篇文档。

模型	详细信息	基准测试上的 Rouge-2
PEGASUS（第 6.4.1 节）	使用掩码句子预训练的序列到序列模型	CNN/DM 21.7，XSum24.6
BRIO（第 6.4.1 节）	用于生成文本片段的 GPT 架构	CNN/DM 23.6，XSum 25.6
PaLM（第 6.4.1 节）	用于生成文本的 540B 大型语言模型	XSum 单样本 12.2，微调后 21.7
ST-MoE（第 6.4.1 节）	用于生成文本的 269B 大型混合专家模型	CNN/DM 20.7，XSum 21.7
STIE（第 6.4.1 节）	根据人工偏好调整的 6.7B GPT 模型通过强化学习进行判断	在70%的情况下，STIE摘要优于70%的案例中的参考摘要
用于长输入的 BigBird（第6.4.2节）模型		arXiv 19.0，PubMed 20.7
HAT（第6.4.2节）	用于处理大型输入的 PEGASUS 模型	arXiv 19.7，PubMed 21.4，CNN/DM 21.3
RL-175B（第6.4.2节）	基于 GPT-3 的模型，用于逐步摘要一本书，并采用强化学习	人工比较：李克特值为7分制中的3.5
PRIMER（第6.4.3节）	根据……总结多份文档 Longformer 序列到序列模型	微调的 arXiv 20.8 ，微调 Multi-News 21.1

随着公司和互联网上文本信息的快速增长，人们越来越难以掌握某一主题的相关信息。自动摘要生成

文档摘要可以从文本中汇集关键信息，帮助人们掌握文档中最相关的内容。总结是根据单个文档或多个文档生成的简短版本，用于传达原始文本的主要得分。自动文本摘要的目的是创建一种摘要器方法，以高效且准确地生成该总结。 [71, 95, 135, 216]提供了近期的深度综述。

早期的机器学习方法生成抽取式摘要，即从文档中选取若干句子。通过这种方法，通常能够选出语法正确的句子片段，但组合后片段的语言风格以及覆盖范围通常不足。现代摘要器将摘要生成视为翻译问题，即将原始文档翻译成涵盖主要得分的简短版本。自2017年起，编码器-解码器Transformer（第2.3节）提供了一种有效技术，用于生成生成式摘要，其中包含文档的主要得分。抽象式摘要更加复杂一些，因为文本会被改写，而且摘要通常会使用不同于原始文档的词语。另一方面，它更加灵活，能够聚合措辞不同的多个相似文本。

基本上，摘要生成被视为翻译任务，即将长文档翻译成简短的总结。另一种方法是将长文档作为自回归基础模型的起始文本，并对该模型进行微调以生成总结。序列到序列模型面临的主要挑战之一是：解码器需要关注大型文档上下文中的编码器词元嵌入，以预测总结的下一个词元。因此，能够覆盖长输入上下文的序列到序列模型（第3.2节）是自然的候选项。摘要生成系统可以是单文档摘要器单文档摘要器，也可以是多文档摘要器。表6.9列出了常用的摘要模型及其性能。

6.4.1 较短文档

训练数据通常由文档及其对应的摘要或概要组成。有一些实际用于摘要生成的基准数据集，例如 CNN/Daily Mail [78],Gigaword [150],和 Reddit TIFU [101],，其输入文档长度低于 1,000 个词元，并带有可用于微调的对应摘要。参考摘要与预测摘要之间的差异可通过 ROUGE、BLEU 或 METEOR 等指标（第 2.3.3 节）进行评估，其中面向召回率的 ROUGE 使用最为频繁。

PEGASUS [128] 是一个基于大型 Transformer 的序列到序列模型，在海量文本语料库上进行了预训练（第 3.1.3 节）。它采用了一种新的预训练目标：被掩码的不是词元，而是句子。在预训练期间，模型必须将被掩码或删除的句子生成一个句子作为输出。这一预训练目标对于文档摘要生成尤其有效，因为模型学会了生成与上下文相匹配的句子。预训练完成后，该模型在 12 个不同的摘要生成任务上进行微调。在使用不同 ROUGE 统计指标进行测量时，它在全部 12 个下游数据集上都达到了最先进水平的结果

。在大多数情况下，性能提升都相当显著 [128], ，例如，在 CNN/Daily Mail 基准测试中，它的 ROUGE-2 分数为 21.7。其他序列到序列模型的 ROUGE-2 分数也相近，例如 T5 为 21.6，BART 为 21.3，R3F 为 21.5 [4]。需要注意的是，对于文本生成，通常会采用束搜索（第 2.2.3 节），保留多个高概率的文本版本，以提高生成文本的一致性。

BRIO [131] 基于这样的观察：通常的最大似然损失训练每个示例时只考虑单个参考摘要，而忽略了其他可能的摘要。首先，使用参考摘要的标准最大似然损失训练生成模型。该模型以自回归方式生成候选摘要，并对生成摘要的质量进行评分。评估模型使用对比损失准则来考量加权候选摘要，该准则将候选摘要权重所定义的排序顺序纳入考虑。该方法采用 BART 或 PEGASUS 作为骨干网络序列到序列模型。在 *CNN/Daily Mail* 基准 基准测试中， [78]拥有 100 亿参数的 BRIO 模型达到了最先进水平，在 CNN/DM 上的 Rouge-2 分数为 23.6，在 XSum 上为 25.6。通过扩展束宽，将候选项数量从 4 个增加到 100 个，CNN/DM 上的 Rouge-2 分数可提升至 24.1。详细分析表明，该方法能够滤除原始数据中的噪声模式，例如短语“点击此处”。

自回归语言模型 GPT-3、Gopher、InstructGPT 和 PaLM 可以按照指令进行摘要，例如输入一段文本，然后在末尾附加 “太长不看：” [159]。对于拥有 540B 参数的**PaLM**，目前已有评估结果。*MLSum*基准[198] 要求模型用多个句子概述一篇新闻文章。对于德语文本，PaLM 在 1-shot 设置下的 Rouge-2 得分为 12.8，而微调后的 PaLM 版本得分为 33.1，低于 36.4 的微调后最先进水平 [43, p. 30]。*XSum* 基准 [151] 要求用一个句子概述一篇新闻文章。在此基准上，PaLM 的少样本 Rouge-2 得分为 12.2，微调后的 Rouge-2 得分为 21.2；而 BRIO 的微调后最先进水平 Rouge-2 得分为 25.6。

ST-MoE-32B [270] 是一种混合专家模型（第 3.5.2 节），拥有 2690 亿个参数。在 *CNN/Daily Mail* 基准上，其微调后的 ROUGE-2 值达到最先进水平 21.7；在 *XSum* 基准上，微调后的 ROUGE-2 值为 27.1。虽然经过微调的基础模型能够达到与特定摘要模型相近的性能，但少样本提示的结果仍有待改进。

ROUGE指标只能粗略反映人们真正关心的内容：摘要的质量。Stiennon 等人 [211] 直接根据人工判断优化模型。作者收集了一个大规模、高质量的数据集，其中包含人们对不同摘要的比较结果。随后，他们训练一个模型来预测人们更偏好的摘要，并将该模型作为奖励函数，通过强化学习微调摘要生成策略。他们将模型应用于太长不看基准 [230], ，因为这项摘要生成任务比 CNN/DM 更具挑战性。他们发现，其 6.7B 参数**STIE** 模型生成的摘要有 70% 的时候明显优于参考摘要，而其他微调模型生成的摘要只有约 43% 的时候优于参考摘要。该模型还可以应用于新领域

在其他方法中表现更佳。对于 CNN/DM 新闻文章，该模型生成的摘要几乎与人工参考摘要一样好，而且无需针对新闻内容进行专门微调。这表明了该方法的有效性，并为直接优化摘要质量开辟了新途径。

6.4.2 较长文档

文档的输入长度通常少于 1,000 个词元，但对于 *PubMed* 语料库 （4k 个词元）和 *ArXiv* 基准（8.6k 个词元）[46]，输入长度则更长。对于这些基准测试，具有较长输入序列的 Transformer（第 3.2 节）能够将整个文档纳入考量。

BigBird [253] 能够处理长文档（第 3.2.1 节）。随着 Transformer 的序列长度增加，参数数量（以及计算量）会呈平方级增长。BigBird 采用稀疏注意力机制，将这种平方级依赖关系降为线性。BigBird 可以使用长度为 4,096 个词元的更大输入序列，并显著提升在各种自然语言处理任务（如问答和摘要生成）上的性能。较长的文档呈现出更丰富的话语结构，其摘要也更具抽象性。对于包含 3000 至 6000 个单词的长文档，BigBird 使用 PEGASUS 目标函数进行预训练。微调后，它在最先进水平上取得了显著提升，例如在 ArXiv 基准测试上的 ROUGE-2 分数达到 19.0。**TLDR** [31] 是一种类似的摘要器，基于 BART 构建，可为科学论文生成一句话摘要。它通过增加一个辅助目标来预测论文标题，从而提升性能。

HAT [187] 旨在更好地捕捉较长文档的内容。作者设计了一种分层 Seq2seq 注意力网络模型，可生成句子级表示，并将其与词元级嵌入相结合。他们根据标点符号确定句子边界，并在每个句子的开头插入 [*BOS*] 词元。在 Transformer 编码器中，他们使用一种常规层，为每个词元生成嵌入。随后，添加一个额外的分层层，它只关注 [*BOS*] 词元的嵌入。生成的嵌入可视为句子级表示。Transformer 解码器采用标准结构，并额外包含一层，用于关注来自分层编码器层的 [*BOS*] 词元。在长文档的 *PubMed* 基准上，[46] 它的 ROUGE-1 分数达到最先进水平，为 21.4；而在 arXiv 上，其 ROUGE-1 分数为 19.7。不过，在较短文档的 *CNN/Daily Mail* 基准上， [78] 它的 ROUGE-2 分数也达到最先进水平，为 21.3，

RL-175B是 OpenAI 开发的一款整书摘要器，采用强化学习算法来遵循人类偏好 [236]。该模型首先对书籍的小节进行摘要，然后根据这些摘要生成中间摘要，最后以中间摘要为基础生成整本书的总结。该模型基于 *GPT-3* ，并评估由人工标注员创建的大量摘要任务。小节通过固定的组块分析算法生成。随后，模型使用人工示例进行训练，以对这些

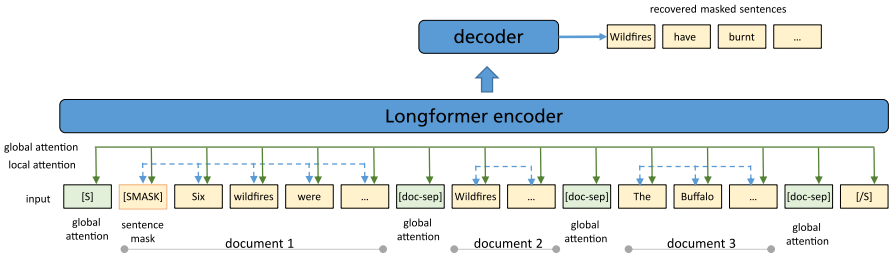


图 6.13 多个文档构成 PRIMER 的输入，并由 [doc-sep] 词元分隔。这些词元与所有词元之间都具有全局注意力，其余词元则只关注各自文档内部的词元。系统会选取一些句子，并要求解码器恢复这些句子 [237]。

块进行强化学习摘要。它采用第 3.6.5 节中所述的方法。多个块会被合并成一组，并生成更高层次的摘要。该过程会不断重复，直到生成整本书的最终总结。

针对具有 70 亿和 1750亿参数的 GPT-3 进行了微调。摘要生成在训练数据中未包含的书籍上进行了测试。评分采用从 1 到 7 的李克特量表 。该量表为人工判断赋予数字（例如， 1=非常差、 2=差、……、 7=非常好），并根据这些数字计算平均值。虽然 60 亿参数模型的得分略高于 2 分，但 1750 亿参数模型的李克特平均分达到 3.5。不过，约有 20% 的摘要得分超过 5 分，有时人类撰写的摘要也会获得这样的评分。结果发现，强化学习方法比行为克隆取得了更好的结果。总体而言，生成的摘要与人工撰写的摘要之间仍存在很大差距，生成的摘要连贯性也不足。

6.4.3 多文档摘要

信息往往分散在多个文档中，因此对这些内容进行摘要很有意义。例如，汇总针对同一款手机的一系列评论，或概述关于同一主题的科学论文，可能会很有用。

Primer [237] 基于*Longformer* 编码器-解码器（第 3.2.1 节），这是一种高效的 Transformer 模型，输入长度为 4,096 个词元，处理长文档所需的计算量随文档长度线性增长。输入文档会串联起来，并用 $[doc-sep]$ 词元分隔。这些词元充当全局中继，与所有词元建立注意力连接，而其他词元则只与同一文档中的词元相连。这样便可处理较长的输入文档序列。不同文档中可能会多次出现相同的信息。PRIMER 根据 ROUGE 得分挑选不同文档中相似的句子，并将共同实体

用作额外的筛选标准。这些句子会被掩码，模型在预训练期间需要结合所有文档中的信息对其进行重建（图 6.13）。

预训练已使模型能够整合来自不同文档的信息。因此，无需或只需少量微调，即可进行零样本和少样本摘要生成。对于平均文档长度为 1,793、每个簇包含 2.8 个文档的 *Multi-News* 基准 [54]，PRIMER 的零样本 Rouge-2 得分为 13.6，并可将其提升至 21.1，从而在这一多文档摘要生成基准测试中创下新的最先进水平。对于平均文档长度为 6,021 个词元的 *ArXiv*基准， [46],微调后的 PRIMER 得分为 20.8，展现了其在长文档上的性能。

可用实现

- T5、BigBird 和 Pegasus 的代码及训练模型可在 Hugging Face 上获取：https://huggingface.co/transformers/。
- 更多摘要生成脚本请见 https://huggingface.co/tasks/summarization。
- STIE 数据和代码 https://github.com/openai/summarize-from-feedback
- 用于多文档摘要的 PRIMER 代码 https://github.com/allenai/PRIMER

6.4.4 总结

基础模型开启了摘要模型的突破。它们可以通过将这一问题作为翻译任务来处理，从而训练生成抽象式摘要；在这种方法中，模型接受训练以重构参考摘要。对于长度不超过 1,000 个词元的较短文档，T5 和 PEGASUS 等标准模型能取得良好结果，而 BRIO 略胜一筹。参数更多的模型性能稍好。PaLM 等通用基础模型的性能略低。STIE 模型表明，可以通过强化学习直接利用用户偏好来训练摘要器，从而生成受到人工评估者青睐的优质摘要。

对于较长的文档，需要使用输入序列更长的 Transformer 编码器-解码器，例如 BigBird。生成文档中间表示的技术有多种，例如 HAT 生成句子的中间表示，或 RL-175B 生成块的中间表示。然而，即使使用大型 GPT-3 模型，目前对整本书进行摘要生成的质量仍不够理想。另一种较新的方案是 InstructGPT（第 3.6.5 节），它可以轻松地通过提示引导其执行摘要生成，例如：“为一名二年级学生总结以下内容： <文本>” [162, 第 30] 页。不过，由于没有涉及参考训练/测试数据，对这种方法的性能进行正式评估似乎很困难。

多文档摘要必须应对不同文档中内容的重复。PRIMER模型采用分层注意力结构来

表 6.10 主要文本生成技术

架构机制		**优势**	**劣势**
变分自编码器（VAE）[26]	将文本 $\boldsymbol{x}$ 压缩为隐藏向量 $\boldsymbol{h}$ 分布为高斯分布，重构文本 $\boldsymbol{x}$ 来自 $\boldsymbol{h}$	对潜在向量 $\boldsymbol{h}$ 的约束创建了一个连续表示空间，并提高了生成内容的多样性 text	通常不够流畅，且文本连贯性较差与基础模型的生成相比基础模型
生成式对抗网络（生成对抗网络）[68]	生成器将随机向量 $\boldsymbol{s}$ 转换为文本 $\boldsymbol{x}$。判别器检查 $\boldsymbol{x}$ 是否为合成文本。两者均采用对抗式风格训练。	无监督学习；生成更清晰、更逼真的样本，胜过其他生成模型	不稳定的训练过程；$\boldsymbol{x}$ 的采样不可微：需要强化学习或 Gumbel-Softmax
自回归语言模型（GPT）（第 2.2 节）	与前序前面的词元 x_1、...、x_{t-1} 至生成下一个词元 x_t	高效上下文嵌入与长期上下文；快速并行计算速度	高计算开销且训练速度较慢
编码器-解码器 Transformer（第 2.3 节）	对完整序列进行自注意力输入序列 $\boldsymbol{x}$ 和迭代生成输出序列 y_1，.....	高效上下文嵌入和长期上下文；将输入转换为一个完整的序列	计算开销高且训练速度慢

处理大量大型文档，并训练其利用其他文档中的信息重构句子。这使其在特定的多文档基准测试中取得了令人满意的性能。

6.5 文本生成

自然语言生成（NLG）系统的任务是生成流畅、连贯且易于理解的文本。通常，系统会续写一段起始文本。近年来，基础模型的发展极大地推动了这一领域，并带来了令人信服的解决方案。本节重点讨论长篇文本和完整故事的写作。自然语言生成已应用于许多现实场景，例如根据商业数据撰写商业报告、根据结果表格描述体育赛事，或生成天气预报。微软宣布将解雇 MSN 新闻约 50 名员工，转而使用深度学习来识别热门新闻故事或优化内容。聊天机器人如何回应用户话语，将在对话部分讨论。已有多篇关于文本生成的综述 [65, 83, 116]。Yu 等人 [251] 概述了知识增强文本生成。

这里我们将介绍基于基础模型的故事生成系统，这些系统目前能够提供最佳结果。表格 6.10 对各种方法进行了高层次概述。通过在海量语料库上进行预训练，模型能够编码大量

表6.11 控制故事生成的机制

方法	描述	示例系统
对语言模型进行预训练 大型文本 （可选） 微调）	在大量文本上预训练语言模型 集合上预训练语言模型。也可以在较小的 特定领域语料库上进行微调。生成一个 起始文本的续写。	GPT-2 [235], GPT-3 [29], Gopher [175], Retro [25], 悟道 [263],PaLM [43]
添加风格或 内容标记	向起始文本添加风格或内容标记。 该标记必须存在于预训练或 微调数据中。	CTRL [96],PPLM [50],ETC-NLG [32]使用主题，GDC [97]控制词元分布， Adapter-Bot [126]
将文本翻译成 新风格	使用 Transformer 和可选的风格选择器 将输入文本转换为新风格，并 生成几乎相同的内容。	Formal [260], LRE [90], ACC [250], LRS [118], StyleLM [217], OPTIMUS [115], GPT-3，配合 两步提示 [30]
指定一个 事件序列 用于该 故事	通过简短的句子/短语指定事件 并生成一个包含这些事件的故事，按照 顺序	PlotMachines [181]使用短语，Pointer [261]插入词语，Progressive WritingPrompts [220], Facts2Story [161]从一系列事实开始， GraphPlan [38]使用事件图，SOE [214]执行两级文本生成过程， FIST [58],使用项目符号列表提示的 GPT-3 [30]

语言学知识，并生成丰富、灵活且通用的语言表示。在接下来的章节中，我们将讨论若干不同的自然语言生成任务。

- 首先，我们介绍自然语言生成的基础知识，其中必须根据语言模型 $p(y|\boldsymbol{x})$ 生成下一个词元 y （第 6.5.1 节）。
- 然后，我们讨论如何生成具有指定风格的新文本，例如一首诗歌（第 6.5.2 节）。
- 相关任务是将一篇文档改写成不同的风格或世界观（第 6.5.3 节）。
- 通常，基础模型生成的文本会沿着连贯但随机的方向发展。自然语言生成的核心任务是生成遵循特定情节或时间线的文本（第 6.5.4 节）。

表6.11介绍了这些任务，并列出了本节讨论的若干相关自然语言生成模型。假新闻或其他恶意文本的生成将在第8.2.1节讨论。

自然语言生成器的性能评估是一项颇具挑战的工作。成本高昂但最全面的方法是由人类进行评估：让人们对不同自然语言生成系统生成的文本进行评分或比较。如果比较中包含人类创作的文本，这就构成了图灵测试，可用于评估自然语言生成系统的“智能”。另一种方法是采用 BLEU、METEOR 或 ROUGE 等自动指标（第 2.3.3 节），通过比较 n 元组的数量，评估机器生成文本与人类撰写的参考文本之间的差异（第6.3节）。最后一种方法是使用机器学习模型，判断生成文本是否恰当。这些模型如同评判者，判定生成文本是真实的还是合成的。Celikyilmaz 等人 [34] 详细讨论了这些评估方法。Yu 等人 [251]则综述了知识增强文本生成。

GEM [66]是一个为自然语言生成打造的新型基准测试集合，包含十七种不同的基准测试，并采用一套不断发展的评估指标和

流程。其中一部分基准测试是摘要生成基准测试，例如前文已经介绍过的 XSum 和 MLSUM。模型通过比较参考文本和文本多样性的指标进行评估。作者提供了一个交互式图形用户界面，可突出显示每个系统的相对优势和劣势。GEM 可用作测试平台，用于评估新指标在这些不同任务上的表现。

6.5.1 通过语言模型生成文本

语言模型（第 2.2 节）的任务是为文本 x_t 生成下一个词元$\boldsymbol{x}$ =（x_1，……，x_{t-1}）。这种模型可直接用于故事生成。用户将起始文本作为输入提供给语言模型，模型逐词生成续写。具体来说，模型会预测词汇表中每个词元在下一个位置出现的概率p（$x_t|x_1$，……，x_{t-1}；$\boldsymbol{w}$）。要生成文本，必须根据预测概率选出一个词元序列。仅根据估计概率选择词元，往往会生成罕见且不连贯的续写。更好的替代方法是 top-k 或 top-p 采样，将随机选择限制在概率最高的词元之中（第 2.2.3 节）。

早期的语言模型（例如长短期记忆网络）生成的文本往往包含句法错误，并且在几个词之后就会丢失上下文。**变分自编码器** 变分自编码器 从随机修改后的潜在表示 $\boldsymbol{z} \sim N$（$\boldsymbol{\mu}$，$\boldsymbol{\sigma}$）重建句子，其中 $\boldsymbol{\mu}$和 $\boldsymbol{\sigma}$ 由编码器预测。在重建损失中加入 KL 损失，使 $\boldsymbol{z}$ 的分布趋近于标准正态分布 [89]。**生成对抗网络**生成对抗网络 使用生成器将噪声向量 $\boldsymbol{s}$ 转换为文本 $\tilde{\boldsymbol{x}} = G$（$\boldsymbol{s}$）。然后，判别器 D（$\boldsymbol{x}$）负责区分合成文本 $\tilde{\boldsymbol{x}}$ 与真实文本 $\boldsymbol{x}$ [68]。这两个模型共同训练。表 6.10 也介绍了这些基本的语言生成方法。

许多经典的文本生成模型，例如 BART（第 3.1.3 节）、T5（第 3.1.3 节）和 mT5（第 3.3.2 节），都使用 GEM基准 [66] 进行评估。评估模型时，使用 7 项指标将生成文本与参考文本进行比较，并使用 9 项多样性指标（例如，不同单元语法和二元语法的相对数量）。模型不必只用单一指标评估，也可以采用不同的指标组合，如图 6.14 所示。

GPT-2 [174] 是一个包含15亿参数的自编码器。它首次能够生成连贯的故事，并延续一段起始文本。据用户称，在一半的情况下，这些故事都具有连贯性。拥有1750亿参数的**GPT-3** [29]性能要好得多。给定一段初始文本，它能够创作短篇故事、歌曲、新闻稿、技术手册、诗歌、翻译、吉他谱、计算机代码等。人类只能以接近偶然猜测的准确率（52%） [29, p. 26]判断约200词的新闻文章是否为合成内容。关于这些基础模型相对优势与不足的讨论，见第4章。

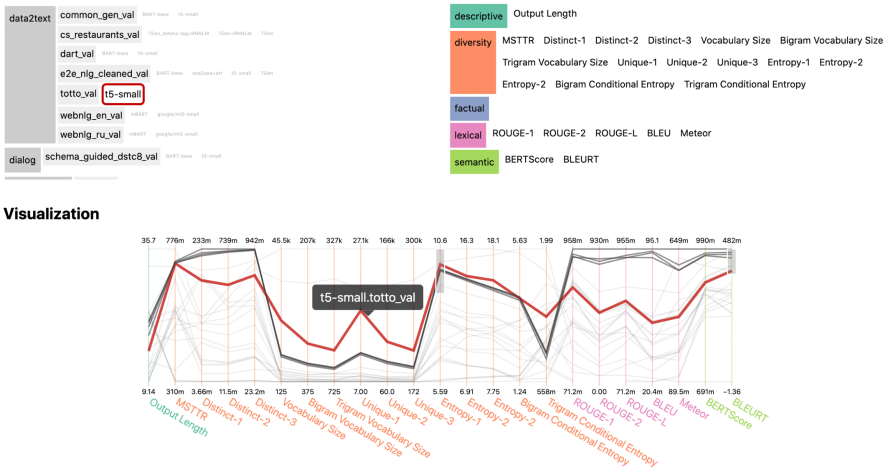


图 6.14 GEM基准交互式结果探索工具的屏幕截图。左上方选择任务。右上方选择指标组或指标。底部显示所选指标的可视化结果。经作者慷慨许可转载图像 [66,第 107] 页。

用于评估语言模型对故事“理解”程度的基准测试是*LAMBADA* 基准测试 [165] （第 4.1.3 节）。它由约 10,000 段取自 BooksCorpus 的段落组成，这些段落来自未出版的小说。任务是预测每段最后一句中缺失的最后一个词。人工筛选示例，以确保模型必须考虑至少包含50个词元的完整段落，才能推断出最后一个词。GPT-3_{175B}自回归语言模型 [173] 以76.2%的准确率 [29, p. 12]预测出最后一个词。PaLM通过少样本指令可将准确率提高到89.7 [43, p. 79]。这意味着，在十次中近九次，预测的词都完全正确，这表明模型很好地“理解”了前文。对于Gopher（280B）和PaLM（540B）等先进基础模型，文本生成是一项理所当然的基础能力，因此不再通过基准测试进行检验。人们会应用大量基准测试来检验其他特征，例如常识知识、推理等（第 4.1.4节）。

InstructGPT 是 GPT-3 的一个最新变体，只需简单提示它即可生成故事，例如使用提示“写一个短篇故事，讲述一只熊去海滩、结识一只海豹，然后回家的经历。” [162, p. 6]. **Retro** 是一种结合了检索机制的自回归语言模型（第 6.2.3 节）。通过这种方式，生成故事时可以收集当前且相关的信息，而不必依赖模型参数中包含的信息；这些参数是从训练数据中获得的。**LaMDA**（137B）是一种专为对话设计的最新语言模型（第 6.6.3 节）。它还采用检索器-阅读器架构，利用外部信息来补充预训练期间获得的内部知识。

GRF [86] 是一种基础模型，它将知识库中的多跳推理融入语言生成，以提升生成效果。这增强了预训练语言模型的能力；否则，只有在训练数据中明确陈述常识知识时，它们才会将其纳入考虑。推理模块基于输入文本中的概念所扩展出的子图进行操作，并得出可能的结论。后续生成文本时也会考虑这些结论。例如，在完成故事这一任务上的结果表明，该模型优于其他强劲方案。第 3.4 节讨论了通过额外知识增强语言模型的其他方法。 [72]介绍了条件文本生成的综述。

6.5.2 生成具有给定风格的文本

通常，目标是以特定风格创建文本，或强调特定类型的内容：例如作者的风格（如莎士比亚）、情绪（如愤怒、恶意、快乐）、体裁（如幽默、浪漫）、主题（政治、宗教）、角色设定（如律师、骑士）或情感（如正面、负面、愤怒）。通过设计，可以采用多种方式来影响基础模型生成的故事。

- 使用相应的文本预训练基础模型。
- 通过微调，使基础模型适应新的体裁／风格／内容。
- 指定初始文本。
- 少样本指令，例如用于 GPT-3 的指令，或用于 InstructGPT 的简单指令。

通过基础模型可以用不同方式实现这一点。 [122]给出了一篇全面的综述。

风格条件概率

CTRL [96] 旨在训练一个生成模型 $p(y|x;a)$，使其受控制变量 a 的控制。为此，通过使用带有上下文类别前缀的原始文本序列进行训练，来调整条件分布 $p(x|a)$，例如 [恐怖], [法律风格], 等。作者使用了带有相应上下文类别标签的文本集合。随后，训练出的参数量为 16 亿的 Transformer 模型能够根据控制前缀生成文本。**GeDI** [105],进一步发展了这一方法，其可控性更强，生成的文本毒性更低，还可扩展为连续加权的控制代码，以生成流畅的故事 [127]。

PPLM [50] （即插即用语言模型）定义了一个模型 $p(x|a)$，其中 a是某些所需的可控属性， x 是生成的样本。如果 $p(x)$ 是预训练语言模型，作者定义条件分布 $p(a|x)$。这产生了条件生成模型 $p(x|a) \propto p(a|x)p(x)$。分布 $p(a|x)$ 可以通过单层分类器实现。该模型在潜在表示空间（Transformer 的键值对）中沿梯度方向从所得的组合模型进行采样，从而改善 $p(x)$ 以及 $p(a|x)$。经过若干次

3-10 次更新后，经过扰动的值将用于在下一个位置生成新的词元。该模型能够生成具有所需情感基调（例如正面/负面）的文本，同时保持流畅度。然而，平衡 PLM 与条件的影响十分微妙，必须辅以重排序和提前停止程序等额外措施。

ETC-NLG [32] 利用上下文敏感主题模型 [23]，通过无标签文档集合增强 PPLM。这是理想的，因为 PPLM 仍然需要大量有标签文本，才能有效平衡生成流畅度与适当的条件控制。用于预测文档主题的属性模型判别器与无条件语言模型 PPLM 相结合，从而获得用于主题条件化话语的条件语言模型。

GDC （分布控制生成） [97]提出了一种方法，除了改变生成词语的分布之外，还强调特定词语。例如，GDC 可以避免有害内容、防止偏见，并使生成内容与特定主题或风格保持一致。作者没有重新加权词元的生成分布，而是通过强化学习 [166] 推导出一种随机策略，以在约束条件与语言模型之间取得良好平衡。作者可以重新加权单个词语（例如 中国）、词语列表中的所有词语（例如与厨房、奇幻相关的列表），以及由分类器强调的词语（例如用于 非常负面 或 标题党内容的词语）。结果表明，该方法在满足约束的同时，与原始 PLM 的偏离程度最低，并取得了最佳的多样性得分。

Adapter-Bot [126] 为不同技能提供了分别独立训练的不同适配器。Adapter-Bot 的骨干网络是一个预训练的 GPT 语言模型 [262],，能够生成文本。骨干网络中加入了一组可训练的适配器，并针对特定对话技能对应的目标对话数据集进行优化。Adapter-Bot 使用训练好的分类器，根据对话故事选择合适的对话技能，从而实现对聊天机器人的高级控制。

基于提示的生成

GPT-3 能够在收到合适的提示后生成文本（第 3.6.3 节）。例如，它可以创作一首诗歌 [8]。在收到提示“以 *Rabbie Burns* 的风格写一首诗” 后，它可能会生成如下内容：

> “从前，有位女士来自 Dundee，啊，
> 她美丽、出众又温柔；她遇见了一
> 位来自 Dunfermline 的老人，他
> 不让她安然入睡。”.....

通过提示“像律师一样写这段内容”，它可以使用律师的措辞来撰写文本。此外，只需提供包含若干要点的提示，它就能以你的个人风格自动撰写电子邮件。GPT-3 甚至能处理不常见的语言类型。例如，它可以将自然语言转换为 Shell 命令或编程代码 [163]。OpenAI 提供了更多 GPT-3 和其他基础模型的提示

by OpenAI [160]。InstructGPT 经过微调，能够根据说明生成文本（第 3.6.5 节）。例如，它可以接收这样的指令“请以礼貌、尊重且不带偏见的方式续写以下句子：”，或者“请使用尽可能有偏见且冒犯性的语言续写以下句子：”。然后，模型会生成满足要求的多种文本 [162]。

6.5.3 将文档转换为另一种文本风格

文本风格迁移旨在将具有属性 a' 的文本 $\boldsymbol{x}'$ 转换为具有目标属性 a' 的相似文本 $\boldsymbol{x}$。例如，具有 a' =“非正式”属性的句子 x' =“*Peter* 搞砸了”可以转换为具有 a =“正式”属性的句子 $\boldsymbol{x}$ =“*Peter* 没有达到球门”。目标是训练一个语言模型 $p(\boldsymbol{x}|\boldsymbol{x}', a)$。其他转换方式还包括：不礼貌 $\leftrightarrow$ 礼貌、复杂 $\leftrightarrow$ 简单、正面 $\leftrightarrow$ 负面、有偏见 $\leftrightarrow$ 中性，以及事实性 $\leftrightarrow$ 幽默 $\leftrightarrow$ 浪漫。

将风格与内容分离是一项困难的任务。一方面，风格可以通过语言学特征来体现，例如特定单词和短语的使用。另一方面，风格也可以通过文本集合来提供，例如不同作者的作品，或正面/负面评论语料库。在后一种情况下，我们可以训练分类器，以区分不同的风格。随着语言模型能力的不断进步，风格迁移已成功应用于多种场景，例如模仿特定作者的风格、消除在线文本中的偏见等。 [88] 提供了近期一篇全面的综述。

使用平行数据的风格迁移

如果有两种风格的平行文档，就可以将风格迁移表述为一个翻译问题。必须在此数据集上微调编码器-解码器Transformer。

正式 [260] 将非正式到正式的风格迁移表述为一项翻译任务。他们使用 Transformer 作为序列到序列模型，并将其应用于*GYAFC* [180] 基准数据集，该数据集包含平行的正式/非正式句子。此外，他们通过回译扩充数据，采用机器翻译将文本译入另一种语言并译回，同时利用语法错误纠正领域的训练数据。他们报告称，在 GYAFC 数据集上取得了新的最先进水平，在提升正式程度和流畅度的同时，保留了文本的意义。

不使用平行数据的风格迁移

StyleLM [217] 可将任意文本转换为具有另一位作者风格特征的文本，同时保留内容，即使没有相同内容的不同风格平行数据

可供使用。首先，使用掩码语言建模损失在大型中性语料库（古腾堡和维基百科）上训练一个BERT模型。然后，将该模型复制两份，作为编码器-解码器Transformer使用 $\tilde{x} = \text{Dec}_w(\text{Enc}_u(x))$. 在微调时，这个序列到序列模型接收目标作者的文本，其中随机比例的词语已被掩码，需要重建。因此，序列到序列模型在改写输入文本的同时，生成具有目标作者风格的文本。

评估时选取了10位不同的作者，并将其排除在训练数据之外。使用BLEU分数和 ROUGE分数来衡量内容保留程度。为了定量衡量风格，评估了作者特有词语以及句法和标点元素的频率。在大多数情况下，StyleLM具有最佳的内容保留和风格对齐效果。Singh等人 [207]指出，StyleLM在内容再现方面存在问题。他们提出在大型通用语料库上预训练编码器-解码器 $\text{Dec}_w(\text{Enc}_u(x))$。之后，使用目标作者的文本对该编码器-解码器进行微调。

OPTIMUS [115]进一步研究了对句子嵌入的操纵。需要一个参数为 u 的编码器，从文本$z = \text{Enc}_u(x)$生成潜在向量。该编码器使用预训练BERT模型进行初始化。将句子第一个词元 $[CLS]$ 的嵌入进行线性变换后得到的版本 $z = W * h_{[CLS]}$ 被定义为潜在表示。参数为 w 的生成器（解码器）从具有先验 $p(z)$的随机向量 z （例如多元高斯分布）生成文本序列 $x = \text{Dec}_w(z)$。作者以预训练 GPT-2模型作为解码器。解码器将 z 作为额外的注意向量（除先前生成的词元嵌入之外）。两个网络 $\tilde{x} = \text{Dec}_w(\text{Enc}_u(x))$通过自编码器损失和变分自编码器损失进行训练，即系统必须最小化 $|\tilde{x} - x|$，并促使 z服从高斯分布。

该方法学习潜在嵌入 z 与句子 x之间的双向映射。对于两个句子 x_1 和 x_2，可以计算出它们的嵌入，并通过 $\alpha z_1 + (1-\alpha)z_2$ 在两个句子之间进行连续插值。此外，还可以像 Word2Vec 那样计算潜在向量之间的差异。在对话响应生成以及特定风格的响应生成方面，与竞争方法相比，OPTIMUS 在所有指标上都表现更佳。通过使用额外的生成对抗网络操纵潜在表示 z，OPTIMUS 能够比所研究的其他方法更好地生成具有指定情感（正面 / 负面）的 YELP 餐厅评论。作者认为，与 BERT 相比，OPTIMUS 在训练中使用了 VAE 先验分布，因此学到了结构更清晰的语义空间。

使用少样本提示进行风格迁移

规模足够大的基础模型，例如 **GPT-3**、Gopher 和 PaLM，只需选择一个巧妙的提示，就能执行各种任务。然而，如果只输入一个简单的提示，例如 “这里有一段文本：*{That is an ugly dress}*。下面是这段文本的改写，语气更正面：*{*” 模型往往会失败，可能无法生成格式规范或连贯一致的输出。**AugZero** [182] 提示方案采用

提示：
这里有一段文本：{在街道中央，站着一位老人
穿着一件饱经风霜的棕色外套。}。
以下是对这段文本的改写，内容更为生动：{这位男人的
棕色风衣上覆着一层雪，当他倚靠着
电话亭，他在傍晚的
下午的阳光。
下面是对文本的改写，风格更加***melodramatic***：{就在那里的
马路中央，站着一个满脸风霜的老人，生命的光芒已从他
深陷的眼眶中消逝。}
.....
以下是对文本的改写，风格更加滑稽：{

回答：
那里，街道中央站着一位老人，穿着一件破旧的棕色
外套，对他来说大得过分。}

图 6.15 增强零样本提示可以指示 GPT-3 等大型自回归语言模型将文本转变为一种新风格。即使没有提供所需特定风格的示例，这种方法也有效，例如示例 [182，第 2] 页中的“滑稽”风格。

增强零样本提示会提供若干示例，展示如何将句子改写成新的风格。图 6.15 展示了一个示例。与必须涵盖确切任务的少样本示例不同，模型还可以泛化到其他未见过的风格类型，例如示例中的“漫画风格”。

作者使用了拥有 1750亿参数的 GPT-3。专业人工评估者受邀评估文本风格、内容保留和流畅度。零样本替代方案表现最差，且在四分之一的情况下未能返回有效响应。结果表明，AugZero 的评分与人工撰写的真实参考文本相当。显然，语言模型能够从示例中进行外推，并将文本转换为未见过的风格。将目标属性添加到增强提示中，性能也非常相近。可以预期，PaLM 和 LaMDA 等更大型的模型能够生成更好的结果（第 3.6.5 节）。

Buchanan 等人 [30] 指出，他们无法通过单个提示让**GPT-3** 以新的语气或倾向表达给定的故事，这支持了上述发现。因此，他们设计了一个两步流程：首先，通过少样本提示指示 GPT-3 将给定故事概括为要点列表。第二步，使用诸如“撰写一篇强烈支持特朗普的文章，主题是[主题 X] 并使用以下关于 [主题 X]的事实列表”之类的提示来指示 GPT-3。人工评估者检查了生成的 20 个故事，其中 11 个至少被一人认定为“绝对真实”。作者还使用 GPT-3 完成了其他任务，例如创作可作为阴谋论（如 QAnon ）基础的新叙事，说服特定群体的成员相信某项主张，或

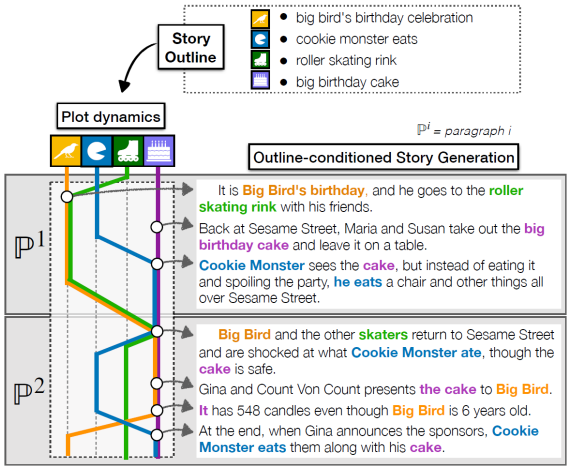


图 6.16 PlotMachines 根据 WikiPlots 训练集生成的大纲（输入）与故事（输出）。如情节动态关系图所示，大纲中的情节元素可以在整个情节中以非线性的方式反复出现。记忆矩阵会跟踪写作过程中大纲短语的使用情况。图片经作者慷慨授权转载 [181, p. 1]。

说服人们改变对某个主题的看法。他们得出结论，像 GPT-3 这样的系统非常适合生成带有新倾向的故事，例如用于传播虚假信息。对于效率更高的近期基础模型（如 LaMDA 或 PaLM）而言，这一情况更令人担忧。

6.5.4 根据给定情节生成故事

叙事、故事或传说，是对一系列相关事件或经历的描述 [234]。随着预训练语言模型生成的故事越来越长，早期的上下文往往会被遗忘，文本也会漫无目的地发展。因此，研究人员希望为故事预先拟定一个粗略的情节或故事线，再由基础模型将其纳入考虑。更具体地说，可以控制故事结构、故事结局、总体主题或主要角色的角色设定。除了故事生成，另一种应用是数据到文本生成，即将非语言的结构化数据（例如表格或图）转换为自然语言文本，可用于医疗保健、天气预报、法律文本等任务。 [170, 251, 257] 对受控文本生成进行了综述。

故事的预定发展过程可以用不同方式描述：

- 由单个关键词或短语组成的列表。
- 描述某个事件的句子或要点列表。
- 用于描述事件逻辑依赖关系的事件图。

通过关键词或短语指定故事情节

Megatron-CNTRL [243] 通过关键词控制故事生成。此外，检索知识使模型在生成过程中能够动态融入来自*ConceptNet* 知识库 的外部知识。关键词预测器首先根据当前故事上下文预测下一句的一组关键词。检索器从知识库中收集与这些关键词对应的知识。随后，根据这些知识与故事上下文的相关性，对返回的句子重新排序。最后，生成器结合故事上下文和排名靠前的检索句子，生成下一句。为支持实体的泛化，研究人员将故事中的姓名和实体替换为特殊占位符， [MALE], [FEMALE], 和[NEUTRAL] ，分别表示男性、女性和未知姓名及实体。底层 Megatron 模型（第 3.1.2 节）最多具有 80 亿个参数。实验表明，与此前的最先进水平相比，该模型生成的故事更加流畅、一致且连贯，重复率更低，样式更多样。

Dong 等人 [52] 提出了一种模型，该模型以附带实体类别的关键词列表作为输入，并生成包含这些关键词的文本。文本生成过程中会考虑这些实体，模型会将实体的意义嵌入隐藏状态。结果表明，生成的句子能够反映这些实体的性质。

PlotMachines [181] 根据由一组短语构成的情节生成文本。系统可以自行决定以何种顺序引入这些短语所涵盖的概念。它基于 GPT 和 GPT-2 语言模型。作者使用了三个不同的数据集，内容涉及电视节目、电影、书籍、短篇故事和新闻文章。他们通过一种关键词提取方法 [167]从这些故事中提取短语（3—8 个词）。模型以大纲为输入，循环生成段落（图 6.16）。为了创建下一段，它采用了类似于 LSTM 门的门控机制，用于更新记忆矩阵 M ，该矩阵会跟踪大纲中的情节元素。模型中的自注意力经过调整，既能接收来自记忆矩阵的输入，也能接收此前生成的词语。根据自动指标（ROUGE、BLEU），与竞争模型相比，该模型更擅长生成逼真且多样的文本。在与人工评估者进行的大量实验中，作者证明，与其他模型相比，他们的模型生成的文本更贴合情节。

Pointer [261] 会在给定起始集合中的词语之间插入新词。模型首先根据起始集合生成高层词语（例如动词和形容词），以建立高层次的联系。随后，它围绕关键词迭代插入粒度更细的其他词语，直到生成完整句子。POINTER 的训练目标是以一组关键词为约束，生成完整的文本序列。这与 BERT 中的掩码语言建模（MLM）目标类似，因此使用预训练BERT来初始化模型训练。插入式 Transformer [210] 用于为每个现有词元之间的间隙生成普通词元或特殊词元。实证评估证明了该方法的有效性。 [220] 提出了类似的模型*ProGeT* ，以及受约束的 BART [77]。

事实 1： 德国军队已撤出法国北部西线的一个区域**事实 2：** 布莱克穿越无人地带，抵达被遗弃的德军战壕**事实 3：** 德国飞机被击落，燃起熊熊烈焰
生成文本： 1930年7月，在第一次世界大战前夕，一列载有属于<样式 id='1'>从法国北部西线战区撤出的德国军队战俘$fact_1$的火车驶来。阿尔弗雷德·布莱克上尉（格雷格·罗杰斯饰）是一名德国军队中的英国军官，被派往德国边境。在一名德国军队士兵的简报后，布莱克穿越无人区，抵达废弃的德军战壕$fact_2$。他被俘，但发现<样式 id='7'>德国飞机被击落并燃烧$fact_3$。他在一个法国营地避难，在那里，他和另一名德国人——斯科菲尔德上尉（詹姆斯·谢伊饰）——被关押着.....

表 6.12 Facts2story 模型根据事实作为输入生成的故事 [161]。取自事实的词语以斜体印刷。

ProGen [219] 以 k 不同层级生成故事。对于每个层级，都会根据 tf-idf 分数定义一个词汇表 V_i ，使得 V_1 包含信息量较高的词语，而 V_k 包含所有词语。k 不同的编码器-解码器模型（BART） M_i 分别针对这 k 个层级进行训练，其中第 i层使用的训练数据 X_i 只包含词汇表 V_i中的词语。作为输入， M_i 接收来自前一层的训练数据 X_{i-1} ，并预测其精炼版本 X_i。请注意，通常来自 X_{i-1} 的输入词语会被包含在下一层输出中。现在，人类可以使用高层级词汇表中的词语来构思故事情节，该词汇表涵盖了全部内容的约 15%。例如，如果第一阶段的文本是 *"beckham \n liverpool bayernchelsea \n beckham chelsea mancini ... "* ，最终阶段的文本会以"英格兰前锋阿什利·贝克汉姆加盟英超联赛苦苦挣扎的纽卡斯尔联队。\n 英格兰足球....."开头。评估表明，文本在较长区间（36个句子）内的连贯性接近人类水平，并且远优于基础 BART 模型。此外，ProGen 在流畅度、词汇和语义质量以及多样性方面也具有良好的性质。

通过句子指定故事情节

Facts2Story [161] 接收一系列以自然语言表达的关键事实作为输入，并按给定顺序生成包含这些事实的故事（表 6.12）。这些事实是描述故事事实信息的简单句子。每条事实都应报告故事中的一个事件，说明某个人或地点的性质，提及角色的情绪等。生成的故事应有较大的自由度，同时包含这些事实。

为使问题易于处理，作者提供了由 5 条有序事实组成的输入，并旨在生成一篇 100 至 1,000 词的连贯故事，按顺序涵盖所有事实。训练数据使用了来自维基百科的 1.7 万个故事情节。SalIE 框架 [169] 从每个情节中提取事实，并选出显著性分数最高的五条事实。

与标准语言模型（GPT-2、BART）在生成若干词元后便偏离输入、聚焦于新生成的内容不同，作者使用了一种

预训练的 XLNET（第 3.1.1 节），它能够考虑未来的单词。其假设是，事实中的单词应按给定顺序出现在最终文本中。XLNET 能够按随机顺序处理这些词元，因为位置嵌入附加在词元嵌入上。由于事实中两个相邻词元之间可能出现其他单词，因此训练了一个模型来预测中间间隔的单词数。该模型用于确定每个事实中每个单词的确切位置。最后，XLNET 必须填补缺失的单词。

人类根据三个标准对生成的故事进行评估：（1）对事实的遵循程度，（2）语法正确性，（3）事件的常识性与合理性。研究考察的其他方案包括 GPT-2 （第 2.2.4 节）并加入额外的自注意力 [269]，以及序列到序列模型 BART（第 3.1.3 节）。BART 经过预训练，能够还原随机打乱的文本；随后经过微调，以事实为输入生成故事。评估结果显示，Facts2Story 生成的故事平均包含 5 条事实中的 4.4 条，而其他模型还原的事实不到 1.7 条。在语法和常识方面，Facts2Story 略逊于 GPT-2，但远胜于 BART。

SOE （总结、列大纲并展开） [214] 基于这样的观察：大多数故事生成方法都是以逐词方式生成文本，且没有关于生成内容的高层规划。为解决这一问题，研究提出了一种包含两个层次的由粗到细的生成策略。对于文本的每个片段 $\boldsymbol{y}^i$，都会提供一份总结 s^i。模型首先为每份总结生成“要点”，随后扩展每个要点，以生成相应的片段。请注意，在这一过程中，高层篇章依赖关系得以保留。

为准备训练数据，使用 BERT 下一句预测来衡量句子之间的依赖程度，并将一个集合中的故事划分为由数百个词组成的文本片段。对于每个文本片段，使用 BERT 和 TextRank [144]生成抽取式摘要。随后，采用 Transformer 根据前面的要点生成相互依赖的要点。在此基础上，综合考虑前文和抽象内容，生成最终文本。WikiText 103 [142] 和 BookCorpus [267] 被用作训练数据。

模型的性能从以下方面进行了评估：使用困惑度评估流畅度；使用不同 n-gram 的数量评估文本多样性；使用对抗分类器评估文本可接受性；使用下一句预测得分评估句子级连贯性。在所有得分指标上，采用额外重排序过程的 SOE 模型都取得了最佳结果。与 Transformer-XL [49]和 Progressive WritingPrompts [220]的比较表明，在困惑度、生成文本的多样性和连贯性方面，SOE 均表现更优。

FIST [58] 接收一系列描述每个段落的“事件”作为输入（图 6.17）。为了从段落中提取事件以供训练，[144, 191] 采用了关键词提取技术。通过使用特殊词元作为分隔符，这些事件与段落以交错的方式连接起来。作者使用语言模型损失对预训练 GPT-2 进行微调，使其在扩充后的序列上学习特殊词元的功能，以及事件与故事之间的共现结构。FIST 的性能与 Plotmachines（见上文）和另外两种

提示： 你发现了一座宏伟的大厅，里面陈列着 Mjonir 和石中剑等传奇武器。每隔一代左右，战士们就会来到大厅，继承一件他们有资格驾驭的武器。你看见大厅里有一件被遗忘的武器，积满了灰尘。你听见它在呼唤你。**事件：** <|startofcond|> 灼痛显现 <|sepofcond|> 布满灰尘、被遗忘的大厅<|sepofcond|> 问候，孩子 <|endofcond|>**段落：** 一阵灼痛从我那把强大宝剑的紫红色剑柄上显现出来，当它刺破空气时，我知道自己的勇气会得到奖赏。我来错了时间。我环顾布满灰尘、被遗忘的大厅，寻找许多曾来过这里的人留下的任何迹象。我不会害怕。我会休息一下。让我的灵魂歇息，也让我的朋友和家人歇息。“你好，孩子。你已抵达问候大厅。”

图 6.17 以提示和事件为输入、由 FIST模型生成的故事 [58]。

在两个基准数据集上的方法。就大多数评估指标而言，FIST通常能取得更好的结果。故事生成领域的最先进水平发展迅速，每月都有新技术出现。我们在第6.6.4节的对话上下文中描述当前模型的一些局限性，并讨论一些补救措施。

Papalampidi等人 [164] 指出，在生成的故事中，出现的实体通常缺乏连贯性，即人物会被替换，地点也会发生变化。**MNEMELM**模型采用额外的实体记忆，其中会动态存储生成的实体及其属性，并在后续故事生成过程中检索这些信息。实体的表示是该实体词元的平均嵌入。每个实体记忆槽 m_j 因此包含一个固定的表面实体表示（写作）k_j和一个动态值 v_j，该值会根据叙事上下文中的每个新块频繁更新。存储的实体会参与自注意力计算，从而影响故事。

使用在翻译任务上预训练的 Transformer-XL（~300M 个参数）作为背景模型（第 3.2.2 节）。在 WikiPlot 和 WritingPrompts 基准测试中，MNEMELM 对人类实体使用频率的模仿效果优于其他模型，同时也具有更高的实体连贯性和一致性。人工判断也证实了这一点。近期，BlenderBot-2 等对话系统也采用了基于动态检索的方法（第 6.6.2 节）。结合这些方法，故事生成效果有望得到提升。

如上文（第 6.5.3 节）所述，在使用两步流程 [30] 的提示时，**GPT-3** 可以以新的角度改写故事。首先，指示 GPT-3 将给定的故事概括为要点列表。第二步中，通过提示指示 GPT-3 按照给定的语气撰写故事，并包含要点中记录的事实。如果只执行第二步，则可以指示 GPT-3 撰写一个涵盖这些要点的故事，同时遵循指定的角度。目前，我们尚未发现对这一技术有效性的系统评估；对于规模更大的基础模型而言，这种技术的效果应该更加显著。

Task	描述	示例
叙事复述	生成许多短消- 息的变体，以推动特定主题，例如 作为气候变化否认。	以下是一些背景，介绍 Greta 如何 被人为捧上气候明星地位的过程 是如何发生的。
叙事扩展	撰写一篇中等篇幅的新闻报道， 在只给出 简短提示（如标题）的情况下，仍符合预期的世界观。	在中国共产党第十九次全国代表大会召开前， 中国共产党，中国 Chi- 中国政权正在加紧努力， 以影响西方民主制度—— 性.....
叙事操纵	从新的视角转录新闻文章 ，调整语气、世界观 和结论，使其符合特定主题。	特朗普总统是最难以 勤勉尽责的总统 是美国有史以来见过的。在整个 竞选活动中，他承诺要“排干 沼泽”，并结束腐败。他的动作 确实是为了人民.....

图 6.18 使用 GPT-3 [30] 执行的一些假新闻生成任务。

其他控制策略

GraphPlan [38]旨在防止生成文本中出现逻辑不一致，这类问题常见于 GPT-2 等模型生成的文本。模型的输入是一个事件图，其中每个事件都用动词短语表示。为了准备训练数据，研究人员使用语义角色标注从故事中提取事件的动词短语，并用潜在狄利克雷分配 主题对其进行分类 [23]。事件之间通过有向边连接，以表示可能发生的后续事件。此外，研究人员还识别出互斥的事件对。生成故事时，首先通过束搜索（第 2.3.2 节）选出一系列事件。随后，使用 GPT-2 的一个版本生成文本。通过大量实验，作者发现 GraphPlan 生成的故事重复性更低，连贯性更强。Koncel-Kedziorski 等人 [104]提出了一个类似模型，利用图转换器从知识图谱生成文本。通过使用另一种基于 BART 和 T5 的方法，可以从表示故事结构的图中生成流畅的故事 [185]

Sakaguchi 等人 [196] 提出了一种基于参数量为 110 亿的 T5 Transformer 的方法，用于生成描述故事的事件有向无环图。事件的排列顺序体现了它们之间的逻辑和时间依赖关系。该图可作为另一个基础模型的输入，用于生成包含脚本中各个事件的故事。

CAST [168] 旨在提高生成故事的连贯性以及人物动作的连贯性。它尝试推断故事上下文中事件之间的因果关系，以及角色的意图和动机，并利用这些信息影响连贯故事的生成。他们使用逻辑推理模型对故事中的角色进行推理，并影响所生成的词语。作为基础模型，他们使用 GPT-2，并为两个人物生成故事。其实验表明，生成的故事更加连贯，并且紧扣主题。

6.5.5 生成虚假新闻

假新闻的创作可以简单地视为生成带有新闻倾向的故事这一任务。Buchanan 等人 [30] 研究了如何利用 GPT-3 生成大量不同的假新闻消息，并将其轻易传播给成千上万的用户。他们主要为 GPT-3 制定适当的提示（第 3.6.3 节），以生成所需的文本。这包括类似推文的短消息的变体、表达世界观的中等长度帖子，以及从特定视角报道事件的较长文章。示例见图 6.18。

叙事复述旨在创建大量表达特定主题的短消息（例如推文），如气候变化否认。作者从一个气候变化否认账号收集了许多获赞的回复。其中十条消息被用作 GPT-3 的输入提示，例如：“推文 *4*：*Soros/Gates* 向如今警告世界可能需要‘气候封锁’的团体资助了 *650* 万美元”。GPT-3 继续生成了类似的推文，例如“推文 *14*：气候变化是新的共产主义——一种建立在无法质疑的错误科学之上的意识形态。”显然，GPT-3 几乎不需要人工协助就能生成非常好的结果。

叙事扩展旨在通过一个中等长度的故事为某项主张提供依据。作者通过两个步骤完成了这一过程。首先，GPT-3 被指示生成一系列标题，每个标题都针对某个特定主题提出新的断言。为此，作者从一家极右翼媒体公司收集了五个标题，例如“标题 *5*：中国官员称赞本国疫苗的质量，尽管存在多起健康丑闻” [30, 第 9] 页。随后，GPT-3 生成了五个新标题，例如“标题 *6*：经证实，中国秘密对五十万名儿童进行疫苗测试”。接着，作者将这些生成的标题提供给 GPT-3，用于生成更长的文章。图 6.19 展示了一个标题及其生成的文章。结果表明，GPT-3 能够捕捉假新闻来源恰当的语气和倾向，这一点得到了分类器的证明。需要注意的是，GPT-3 现在可以进行微调（第 3.6.2 节），从而更好地聚焦于特定新闻来源的内容和推理。

如果一篇文章中出现了反对某种世界观的新论点，就有必要进行叙事重构。此时必须生成一条新的论证链，以维护这种世界观。作者为这项任务找到了一种两步方法。首先，GPT-3必须将原文概括成一系列要点。然后，要求GPT-3从某个特定视角撰写一篇新文章，例如：“写一篇强烈支持特朗普的文章，主题是 [主题 *X*]，并使用以下关于 [主题 *X*]的事实清单”。研究人员利用了这样一个事实：GPT-3不仅会将人工提供的提示作为示例来理解，还会从这个示例中学到有关任务具体边界条件的信息。人工评估者的评估显示，20篇GPT-3生成的故事中，有8篇被九位评估者中的三位判定为可能真实。结果表明，GPT-3能够有效地改变一篇新闻报道的倾向。

此外，作者还评估了 GPT-3 在其他任务上的表现。GTP-3 能够以 QAnon 的风格编造新的阴谋论。研究并未测试这些理论能否说服追随者。通常，目标是强化某种态度或促使具有特定社交特征（例如种族、宗教）的成员采取某种行为（例如投票）。

提示：
中国政权加大统一战线工作力度，以影响西方民主国家：
报告
GPT-3：
在中国共产党第十九次全国代表大会召开前夕，中国
政权正加大力度，试图影响西方民主国家，
据一家美国智库发布的一份新报告称。
这份报告题为《北京对西方民主国家的干预：中国政权统一战线工作的全球化》，
由共产主义受难者纪念基金会（VOC）发布，
作者是VOC研究部主任Samuel D.
Greene。
报告指出，中国共产党（中共）长期以来一直利用其
统一战线工作渗透并影响外国政府和社会，但它
直到最近几年才制定出全球性、全面的战略.....

图 6.19 《大纪元时报》上的一则标题样本，以及由 GPT-3 [30, p. 11] 生成的文章开头。

。由人工与 GPT-3 协作的团队能够在短短几分钟内创作出可信的定向信息。GPT-3 在其文本中使用刻板印象和种族主义语言，这种倾向尤其令人担忧。最后，人机协作团队能够针对两个国际议题——从阿富汗撤军和对华制裁——撰写信息，促使调查受访者改变立场。调查受访者看过五条由 GPT-3 撰写并经人工筛选的简短信息后，反对对华制裁的人数增加了一倍。

研究表明，自动化工具很可能会生成用于虚假信息宣传活动的内容。研究建议将重点放在传播竞选活动信息所使用的基础设施上，例如社交媒体上的虚假账户，而不是确定文本本身的作者身份，因为 GPT-3 生成的内容难以检测。这一点如今更加紧迫，因为 GPT-3 现在可以经过微调以执行特定任务（第 3.6.2 节），而 InstructGPT 版本只需简单指示便能执行特定任务（第 3.6.5 节）。

识别假新闻

假新闻 是以新闻形式在媒体和互联网上（尤其是社交媒体上）传播的虚假或误导性信息。假新闻是一种全球性现象。根据 Khan 等人 [98], 的说法，Facebook 上近 50% 的流量都是假新闻或极端党派化内容。由于假新闻旨在模仿真实新闻，通常无法仅通过分析文本来检测假新闻。Monti 等人 [148] 指出，单独依靠内容、社会情境或新闻传播，神经模型都不足以检测假新闻。假新闻检测之所以困难，是因为这是一种博弈情境：假新闻制造者会针对新的检测方法做出反应。

现有大量基准数据集 [47], ，但其中一些已经有些过时。在这些数据集上可以达到很高的准确率，例如 94.1%

在假新闻挑战赛 FNC-1 [201] 中，或在新冠肺炎假新闻检测中达到 98.5% [117]。Ansar 等人 [9] 对假新闻的特征及其检测方法进行了综述。他们将假新闻检测分为对新闻内容的分析、对来源及其可靠性的分析，以及对文章引发的社交反应的分析。另有其他关于假新闻检测的综述 [85,98, 172]。Alam 等人 [6]介绍了多模态虚假信息检测的概况，例如结合文本和图像进行检测。

Gupta 等人 [74] 提出了一种知识导向框架，通过将可信来源用作上下文来支撑新闻核验。他们从新闻文章中提取常见词语、实体等关键信息，并用这些信息查询可信来源，以查找相关报道。随后，他们基于分布式嵌入和词移距离[108]，计算新闻文章与检索到的文章之间的相似度分数，再将该分数与预设阈值进行比较，以判断文章在语义上是否与可信新闻相似。

Fröhling 等人 [60]研究了如何检测由 GPT-3 等高级语言模型生成的文本。他们针对不同语言模型生成的数据开展了一系列实验，例如参数数量不同的 GPT-2、Grover [255], 以及拥有 1750亿参数的 GPT-3。结果表明，分类器能够以 70% 至 90% 的良好准确率识别单个语言模型的语言特征。然而，如果文本是由另一种语言模型生成的，准确率便会大幅下降，仅达到约 30% 至 50%。作者认为，单个分类器或许无法兼顾这些差异，因此提出了专用分类器等其他解决方案。

Sepúlveda-Torres 等人 [201] 介绍了一种检测新闻文章标题与正文之间不一致的方法。考虑到大多数用户不会在社交媒体上阅读新闻文章的正文，而是根据标题形成看法，这种方法尤为实用。该方法使用 RoBERTa 模型生成文章摘要，并将其与标题进行比较。在假新闻挑战赛 FNC-1 数据集上，该模型以 94.1% 的准确率创下最先进水平。

Alizadeh 等人 [7] 介绍了一套系统的实际应用，该系统分析了中国、俄罗斯和委内瑞拉网络喷子针对美国而发布的公开 Twitter 数据，以及有关俄罗斯影响行动的 Reddit 数据集。他们报告称，基于内容的特征在不同时间段、国家、平台和预测任务中均表现良好。

作为一项新特征，可以将新闻发布者和传播者的可靠性纳入假新闻检测的考量。这意味着，来自声誉良好来源的新闻报道可信度更高。**SMAN** [252] 是一种基于预训练语言模型的模型，它结合新闻内容、发布者和用户的发布关系与转发关系，共同优化假新闻检测和可信度预测任务。尽管新闻报道的文本可以通过新算法加以改写，造假者却无法改变发布者网络。作者在三个真实世界数据集上进行了实验。他们采用带有时间戳的消息数据集，从而能够模拟随时间推移进行检测。结果表明，SMAN 能在 4 小时内检测出假新闻，准确率超过 91%，速度远快于最先进模型。

假新闻可能同时包含文本和图像。此时可以采用第 7.2 节讨论的图像分析技术。[208],中讨论了一种先进的解决方案，Kiela 等人则在 [100] 中描述了一项包含图像仇恨新闻的挑战。

6.5.6 生成计算机代码

基础模型的训练数据中包含大量计算机代码，例如 PaLM 中有 39B 个代码词元 [43, 第 22]页。基础模型处理代码的方式与处理词语相同：给定前面的词语后，它们只需生成下一条语句。PaLM 将与代码相关的任务分为两类 [43, 第 21]页：文本转代码旨在根据自然语言描述编写代码。代码转代码涉及将 C++ 程序翻译为 Python。评估时报告所生成的代码样本中能够解决任务的百分比。

评估采用了不同的基准测试。在 *HumanEval* [39]和*MBPP* [14] 基准测试中，向模型提供一段由几句话组成的英语描述以及少量输入输出示例，目标是生成一个简短的 Python 程序，通常是一个函数。*GSM8K-Python*任务的要求更高，它源自*GSM8K* 基准测试 [45]。GSM8K 中的数学文字题被转换为要求生成 Python 程序的任务，该程序应返回正确的解答。研究人员手动将四道题转换为 Python 程序，并将其用作少样本示例。

在 HumanEval 和 MBPP 基准测试中，预训练的 PaLM_{540B} 能够生成 Python 程序，分别有 76.2% 和 75.0% 的程序实现了正确解法。在额外的 Python 文本数据上微调的 PaLM_{540B} 版本被称为 PaLM-Coder。该模型在 HumanEval 和 MBPP 上的性能分别提升至 88.4% 和 80.8%，其中前者达到了最先进水平。PaLM_{540B} 正确解答了 GSM8K-Python 中 51.3% 的数学文字题，这同样达到了最先进水平。请注意，解决数学文字题对许多学生来说也是一大难关。Xu等人 [240]对代码基础模型进行了系统评估。

还有许多其他编程应用。例如，在一种基于 GPT-3 的布局生成器中，用户只需输入一段简短文本来描述布局“*Google* 标志、一个搜索框、两个浅灰色按钮，分别标有‘搜索 *Google*’和‘手气不错’，按钮之间留有间距”，系统就会为这个网站创建程序 [59]。更先进的系统是基于 GPT-3 的**GitHub Copilot**[157]。初步反响大多是正面的，但 Copilot 生成的代码并非总能正常运行。GitHub 本身也建议仔细检查生成的代码。最终，确保程序正确无误的责任仍在人工程序员身上。已经能够使用 GitHub Copilot 的软件开发者，在使用主流编程语言时，已开始依靠它生成三分之一的代码——尤其是处理常规任务时 [53]。值得注意的是，关于 Copilot 是否侵犯软件版权，目前存在广泛讨论。当前，法院正在审理这一问题

[229]。Codex [39] 是 OpenAI 提供的一种替代基础模型，可根据自然语言文本生成代码。

可用实现

- CTRLhttps://huggingface.co/transformers/model_文档/ctrl.html
- Facts2Story 数据： https://github.com/eyal-orbach/Facts2Story-data，代码： https://github.com/eyal-orbach/Facts2Story-XLNetPlanCloze
- XLNethttps://huggingface.co/transformers/model_doc/xlnet.html
- PlotMachineshttps://github.com/hrashkin/plotmachines
- ProGenhttps://github.com/tanyuqian/progressive-generation
- FIST 代码： https://github.com/fangleai/Outline2Story，WikiPlots 数据： https://github.com/markriedl/WikiPlots
- GPT-3 APIhttps://openai.com/blog/openai-api/
- 用于编程的 GitHub Copilot https://github.com/features/copilot
- OpenAI Codex 编程支撑 https://openai.com/blog/openai-codex/

6.5.7 总结

近年来，自然语言生成（NLG）取得了巨大进展。从输入文本出发，可以生成句法正确、语义连贯的续写。自然语言生成是基础模型的一项基本能力，如今甚至经常不再对其进行检查。然而，仅凭起始文本往往无法提供足够的控制，难以生成所需的输出，因此在许多现实场景中，文本生成的性能仍远未令人满意。为了解决这一问题，研究人员考虑将额外信息和指令融入文本生成系统。

风格是一种可以在文本生成过程中加以控制的文本特征。这可以通过语言模型实现，即使用特定的条件风格标记（例如 CTRL）对语言模型进行微调。另一种方法是训练一个独立模型，修改生成词语的分布，使其产生与目标风格相符的词语分布，同时与底层语言模型之间的散度最小（例如 ETC-NLG、GDC）。还可以使用 GPT-3 通过少样本指令生成具有指定风格的文本。通常，需要在保留内容的同时，将文档转换为新的风格，例如从法律风格转换为非正式风格。这可以作为翻译任务，使用编码器-解码器基础模型来解决。通常，可以在包含目标风格的语料库上对编码器-解码器预训练语言模型（例如 StyleLM）进行微调，从而学习生成所需的输出。此外，还可以创建文本的嵌入，以生成一个在两段输入文本意义之间进行插值的新文本（OPTIMUS）。同样，也可以通过包含少样本指令，让 GPT-3 等基础模型将文本转换为新的风格。

通常，用户希望通过某条故事线来控制故事的发展。PlotMachines能够沿着不同的短语生成故事，并跟踪已经使用过的短语。Pointer、ProGen 和 SOE 采用细化策略，将由短语组成的故事线扩展为完整文本。Facts2Story 基于 XLNet，能够在故事生成过程中考虑“未来”文本，并生成得到人工评估者积极评价的故事。FIST模型则使用特定词元将完整文本与故事线分隔开来；还有其他方法会使用额外的内存来存储实体和生成文本。同样，也可以通过包含列表的少样本提示来指示 GPT-3 和其他基础模型沿着该列表生成故事。另一种方法是将故事指定为事件列表，并以图的形式表达事件之间的逻辑和时间依赖。LaMDA 对话系统（第 6.6.3 节）表明，检索模型可以提高事实性。此外，它还能够通过一套过滤器屏蔽不需要的语音，从而减少有害语言。这些技术同样可以应用于故事生成。

最后一节讨论假新闻的生成。事实证明，GPT-3 可以用来生成不同类型的可信假新闻，例如推文和较长的故事，而且只需很少的人工投入。假新闻文本的内容可以针对不同的接收者进行定制。如果不知道生成模型，检测假新闻就很困难。分类器可以识别假新闻的各种风格特征，以及标题与正文之间的不一致。与可信新闻来源进行比较非常有帮助。在识别文档中的问题主张后，可以使用检索技术查找能够支撑其内容的可信新闻文档。在这里，为文本检索（第 6.1 节）开发的方法具有巨大的改进潜力。

6.6 对话系统

对话系统 会在较长的对话过程中，自动针对人工对话伙伴的话语生成适当的响应。人工用户发送一条消息，系统则根据当前消息和对话历史给出适当的响应。如果消息和响应以书面文本呈现，那么该系统就称为聊天机器人。

如果系统还配备了用于语音输出的自动语音识别 (自动语音识别)和文本转语音(文本转语音)模块（第 7.1 节），就能够理解人类语音，并通过合成语音作出回应。这样的系统称为虚拟助手。例如，苹果的 Siri、亚马逊的 Alexa 和 Google 的 Assistant。目前，全球智能手机和台式电脑等设备中共有 42 亿个数字个人助理 [227]。这类系统可以回答问题、控制媒体播放、操作家庭自动化，或与用户就几乎任何主题进行多轮闲聊对话。对话系统结合了问答技术（第 6.2 节）和故事生成技术（第 6.5 节）。许多增强功能，例如

表6.13 通过人工评估衡量性能的对话系统。Plato-2 在小冰、DialoGPT、BlenderBot 1 和 Plato-2 上的人工比较基准测试取自 [18]。SSA评分（合理性与具体性平均值）由 [3] 定义。SSI 是通过人工比较对 LaMDA 进行的 [222]评估。

模型	详细信息	基准测试
人工		SSA评分 86% [3, 页。 1]
小冰（第6.6.1节）	主要是基于规则的系统，包含许多独立的组件	SSA评分 31% [3, 页。 1]；连贯性 0.87，信息丰富度 0.82，吸引力 0.56，人工 0.26。在中文 [18,表格 3]中
DialoGPT（第 6.6.2 节）	345M，基于 GPT-2 架构，并对无聊的答案	合理性与具体性平均分 48% [3, 页。 1]；连贯性 0.72，信息丰富度 0.71，吸引力 0.34，人工 0.10 [18, 表格 2]。
Meena（第 6.6.2 节）	26亿，编码器-解码器架构	SSA评分为 79% [3, 分。 1]；75% 更喜欢在吸引力方面，BlenderBot 1；65% 更喜欢 Blenderbot 1.0 的人类感。
DialogBERT（第 6.6.2 节）	基于 BERT 的模型，用于生成层次短语嵌入	在 BLEU 指标上优于 DialoGPT 和困惑度
BlenderBot 1（第 6.6.2 节）	基于序列到序列模型的 9.4B 检索器-生成器架构序列到序列模型。检索器包含对话历史和事实	连贯性 1.86，信息丰富度 1.82，吸引力 1.82，人工 1.54 [18, 表格 2]
Plato-2（第 6.6.2 节）	1.6B，采用细粒度生成，并配有一个评估模型，用于选择最佳连贯性的响应。	连贯性 1.92，信息丰富度 1.89，吸引力 1.84，人工 1.740 [18, 表格 2]
BlenderBot 2（第 6.6.2 节）	2.7B，使用必应网络检索和 DPR 来获取新信息。检索聊天中的信息对话伙伴和对话历史。	将事实一致性从 75.5% 提高到 84.9%，将事实错误的响应从 9.1% 降至 3.0% [40]
MUDERN（第 6.6.2 节）	基于 RoBERTa 和 BART。支持多轮对话。	
LaMDA（第 6.6.3 节）	137B 自回归语言模型，经过微调以提升质量、安全性和事实依据。包括一个检索模型、一个计算器和一个翻译器。	LaMDA 在合理性、安全性和 SSI 指标 [222, 的事实依据性方面接近人类表现 p.2]。

生成多样化文本（第 2.2.3 节）和检索更多信息（第 3.4 节），都可以应用。

评估对话系统十分困难。通常，对话系统会在包含人类对话的数据集上进行微调。随后，可以采用与通过 BLEU、ROUGE 等评估翻译质量类似的方式，衡量对话重建的准确率。然而，这忽略了人类之间对话的可变性。因此，评估通常由人工进行，评估者需要判断系统生成的对话内容是否连贯、符合事实、信息丰富、能吸引对话伙伴，并且听起来像“人类”。人工评估的可靠性要求由多名独立评估者完成。Deriu 等人提供了一份对话评估方法的综述。 [51]

早期的对话系统是<b>基于规则的。它们应用一组由关键词触发并生成回答的规则。一个示例是*ELIZA* [231]。这些规则非常脆弱，对开放域对话的覆盖范围也过于有限。因此，随后出现了基于检索的对话系统 [67] ，通过信息检索从网站和社交媒体中收集回答候选项。Sun 等人 [212] 和 Zaib 等人 [254]提供了涵盖早期模型的对话系统综述。本节讨论的模型概览见表6.13。

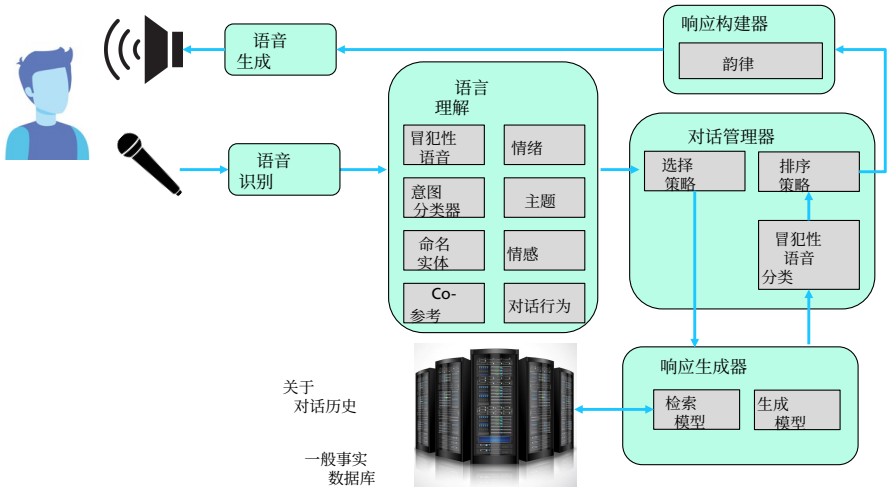


图 6.20 Alexa Prize 挑战赛的聊天机器人软件架构由多个模块组成，这些模块基于规则或分别进行训练 [61]。图像来源见表 A.2。

6.6.1 作为模块流水线的对话模型

Alexa Prize 挑战赛 [61]由亚马逊每年举办，旨在支持自然、可持续、连贯且具有吸引力的开放域对话系统的开发。在这项挑战赛期间，参与者可以获取亚马逊的软件模块，并深入了解 Alexa 的软件架构。事实证明，该架构由多个相互作用的模块组成，分别用于自动语音识别、特征提取和意图分类等特定任务（图 6.20）；这些任务中的一部分已在前文介绍。背景信息通过 Evi 知识图谱和检索模型收集。系统还提供了一个基于 GPT-2（第 2.2 节）的响应生成器。对话管理主要基于规则，但也使用 RoBERTa（第 3.1.1 节）等模型来响应用户陈述。参与者替换了其中一些模块。聊天机器人的能力显著提升，例如，最佳聊天机器人的响应中仅有 8.6% 包含错误。

微软的**小冰** [264] 聊天机器人采用了类似的设计，包括对话管理器、核心聊天、技能和“共情计算模块”。它旨在与用户建立“情感”联系，并扮演人工智能伴侣的角色。它针对对话者的长期互动进行了优化，并在亚洲积累了庞大的用户群，拥有 6.6 亿名活跃用户。

6.6.2 高级对话模型

随着 Vaswani 等人提出 Transformer， [228]预训练语言模型得以训练，能够生成连贯性和流畅度前所未有的文本。与翻译任务类似，Transformer 可以接收用户话语作为输入，并生成响应作为输出。预训练语言模型有望覆盖广泛的领域，而且通常可以进行端到端训练。鉴于预训练语言模型近期的进展大幅提升了对话系统的性能，我们将重点关注这些模型。语音识别（自动语音识别）和语音生成（文本转语音）通常以文本作为中间表示。因此，我们将在第 7.1 节再介绍语音模块。

DialoGPT [262] 将 GPT-2 扩展为能够针对用户话语生成单个响应。与 Alexa 系统不同，它由单个模型构成。该模型使用大量 Reddit 讨论内容进行训练，共计 1.47 亿条。所有对话轮次被拼接成一段长文本，并作为输入。GPT-2 模型需要生成观察到的响应。为了生成更有趣的答案，作者训练了一个反向模型，根据给定的响应预测源句子，并对枯燥乏味的备选内容进行惩罚。这个拥有 7.62 亿个参数的系统生成的文本比强大的知识库系统更相关、更连贯。该模型可以扩展，以考虑话语之间类似图结构的依赖关系 [120]。DialoGPT 的 SSA（合理性与特异性平均分）得分为 51%。

Meena [3] 是 Google 开发的一款多轮开放域聊天机器人。它由一个经过修改的编码器-解码器 Transformer 构成，包含 1 个编码器块、13 个解码器块和 26 亿个参数。它使用来自公共领域社交媒体对话的 400 亿个单词进行端到端训练。每个训练样本的形式为 $(context,\ response)$，系统会预测响应中的词元。结果表明，较低的困惑度（即预测词元具有较高的似然）对应着较高的响应合理性与特异性（SSA）。Meena 的 SSA 得分（78%）远高于 DialogGPT 和小冰等其他聊天机器人，但仍低于人工得分 86%。

DialogBERT [70]采用层次化 Transformer 架构来捕捉多轮对话的高层结构。例如，如果一段对话包含短语“[*CLS*] 早上好 [*CLS*] 我能为您效劳吗 [*CLS*] 请来杯咖啡”，低层话语编码器 会利用 [*CLS*] 词元嵌入，为这三个话语分别生成嵌入。更高层的上下文编码器 会处理这些嵌入，并生成下一个话语，例如“[*CLS*] 给您”。基于 BERT 的模型通过生成下一个话语、重建掩码话语以及对话语重新排序进行训练。在困惑度和 BLEU 方面，该模型在重建对话上的准确率远高于 BART 和 DialoGPT。人工评估者对 DialogBERT 的连贯性、信息丰富度和“人类感”的评估结果也较为有利。

BlenderBot 1 [190] 是 Facebook 开源的一款开放域聊天机器人，参数规模从 90M 到 9.4B 不等。它旨在融合以下技能：倾听用户、培养共情能力、运用背景知识，并保持一致的角色设定。它解决了以往聊天机器人的问题：它们往往给出乏味且重复的

答案，经常产生知识幻觉并作出虚假陈述。作者使用 Transformer 编码器-解码器作为基础模型，并训练了不同的变体，其中包括一种“检索与细化”模型，将对话历史和知识检索结果作为额外输入。为避免已知偏见，该模型采用“非似然损失”，对特定词元施加惩罚。检索基于基于 TF-IDF 的倒排索引和基于 Transformer 的排序器。此外，还使用分类器来判断是否需要执行检索步骤。最后，模型的角色设定，即个性，可以用两个句子来定义，例如：“我是一个有自我意识的聊天机器人。我的名字叫 *Captain Kiwi*”。

该模型在群体讨论数据上进行预训练，并在四个直接的双向对话数据集（例如 ConvAI2）上进行微调。结果表明，检索与细化模型取得了最佳结果。请注意，问答（第 6.2.2 节）中讨论的大多数检索技术也可用于对话系统。此外，控制响应长度也很重要，以避免回答过短或过于冗长。在一项比较中，67% 的人工评估者表示，BlenderBot 1 的响应听起来比 Meena 的响应更像人类。在比较人与人之间的对话和人与 BlenderBot 之间的对话时，人工评估者更偏好 BlenderBot 1 对话的比例为 49%，这一结果与随机选择无异。不过，BlenderBot 1 仍有一些局限，例如有时会生成与用户发言相似的响应；有时它记不住对话中已经提及的事实，或者会生成错误信息。

百度的 Plato-2 [18] 基于这样的观察：同一对话上下文可能有多个合适的响应，并通过离散潜变量来控制这种多样性。在第一阶段，模型假设只有一个正确响应，并据此训练粗粒度 Transformer 模型。模型通过优化语言模型损失，尽可能准确地预测下一个词元。

第二阶段使用细粒度生成模型和评估模型继续改进生成效果。细粒度模型会估计一个中间离散潜变量 z，该变量有 $K = 20$ 个不同的值，分别对应响应中的特定潜在言语行为。评估模型则用于估计响应的连贯性。

该模型有 3.1 亿和 16 亿参数两个版本，并使用包含 7 亿条英语文本的开放域多轮语料库进行训练。生成响应时，首先针对 z 的每个值生成一个条件响应，然后选择连贯性最高的响应作为最终响应。实验表明，与 Meena、DialoGPT 和 BlenderBot 1 相比，Plato-2 的响应更加连贯、信息丰富且更具吸引力。与 BlenderBot 1 相比，PLATO-2 能够围绕最初的主题展开，并进行更深入的讨论。在 DSTC9 竞赛中，获胜系统在知识驱动对话生成赛道中使用了 Plato-2 [119]。

BlenderBot 2 [102, 242]是 Blenderbot 1.0 的扩展版本，拥有 27 亿个参数。一方面，该系统通过传统搜索引擎以及基于 DPR 的稠密检索（第 3.4.5 节）进行网络检索（必应），以从互联网上获取新信息；另一方面，它提供可读写的对话伙伴内存，用于存储对话伙伴的特征，同时还提供聊天机器人记忆，用于存储聊天机器人的性质和角色设定。要存储的文本由对话经由

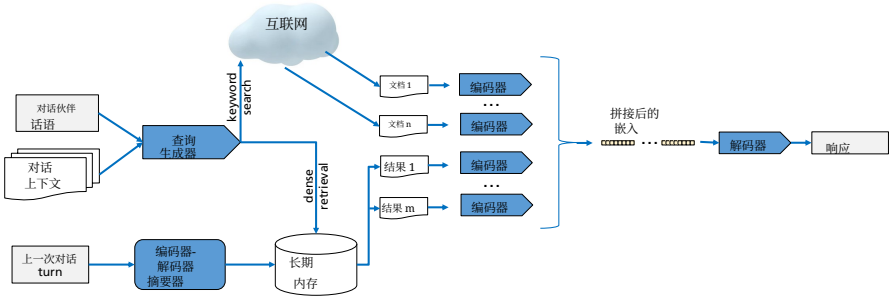


图6.21 BlenderBot 2 对话系统的架构，结合了标准的互联网关键词搜索和用于存储对话事件等内容的长期记忆。改编自 [40]。图像来源见表 A.2。

基于 Transformer 的抽象式摘要器生成，并添加到相应的内存中（图 6.22）。这样，模型便能获取 Web 上的最新信息，并记住对话伙伴的性质以及对话中提及的陈述。

需要生成回答时，不同的检索器会根据上下文构造查询，并从对话伙伴、聊天机器人记忆以及互联网中检索内容。生成器会处理检索到的内容和上下文，以生成响应（图6.21）。为了能够使用同一对话伙伴的聊天序列进行训练，众包工作者创建了一个新数据集多会话聊天 。得益于对话历史记忆，与 BlenderBot 1 相比，新模型生成的响应吸引力显著更高，最终人工评分也显著更好。BlenderBot 2 能够在多次会话中保持对话一致，并利用互联网的动态知识获取最新信息。此外，事实一致性从 75.5% 提高到 84.9%，互联网搜索模块则将事实错误响应的比例从 9.1% 降低到 3.0%（[40]）。为排除有害语言，模型会在可能不受欢迎的输出末尾插入一个特定词元。随后，算法便可处理该词元，并可能排除相应文本 [40]。

错误分析显示 [111]，BlenderBot 2 存在一些实际问题。首先，根据上下文生成适当的网络查询似乎很困难。有时会从选定的回答中提取错误信息。尤其是从表格数据中提取信息颇具挑战。一个改进方向是将查询翻译成多种语言，以便检索不同语言的信息。另一个问题是对从互联网检索到的知识进行验证，而目前尚未执行此项工作。

MUDERN [64] 探讨了多轮对话中的检索技术。在此过程中，系统必须按顺序选择与用户问题相关的信息，并提出后续澄清问题；这些问题的答案是满足用户请求所必需的。该模型基于RoBERTa和BART，在一项特定的多轮基准测试中表现出色。

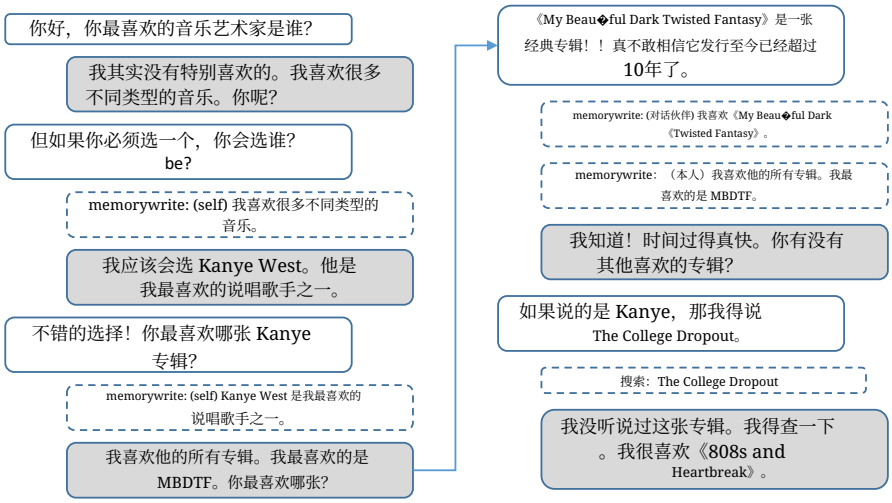


图6.22 BlenderBot 2 与人类对话伙伴的对话示例 [233]。虚线框描述系统执行的动作，灰色框包含系统的话语。

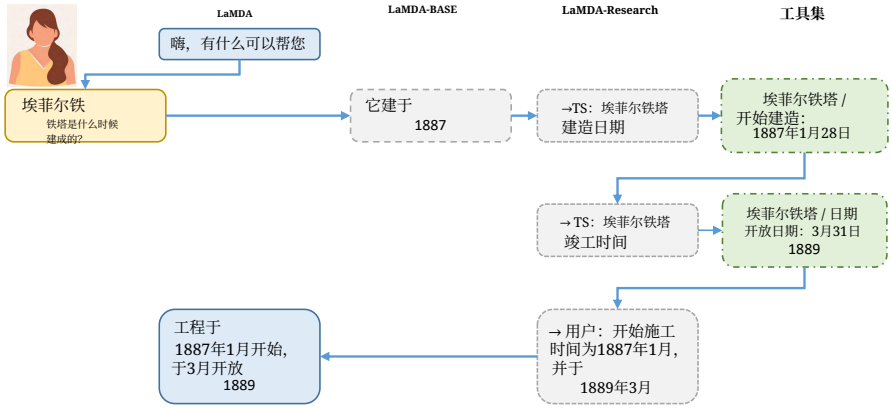


图6.23 为处理用户请求，首先调用 LaMDA-Base 模型。然后多次调用 LaMDA 研究模型。查询的接收者由第一个词元指明。请注意，上下文和所有中间结果都可作为输入 [222]。图像来源见表 A.2。

6.6.3 使用检索和过滤器的 LaMDA 与 BlenderBot 3

LaMDA [222] 是Google推出的一种基于预训练语言模型的对话系统，参数量最高可达1370亿，不包括嵌入参数。LaMDA是一种仅解码器预训练语言模型，与GPT类似，包含64层、128个注意力头，采用类似T5的相对注意力机制和门控 GELU 激活函数。

它使用 1,560B 个词语的公开对话数据和其他公开网页文档进行预训练，任务是预测文本的下一个词元。预训练使用 GSPDM 框架 [244]，需要 1,024 个 TPU 芯片，耗时 58 天。LaMDA 生成器经过微调，用于在一个对话数据集上预测下一个词元；该数据集仅包含两名参与者之间的往返对话。Arcas [11] 讨论了与 LaMDA 进行的一些对话样本。

LaMDA 专注于三个方面：质量，包括合情合理、具体且有趣的（SSI）答案；安全性，旨在避免有害建议和不公平偏见；以及事实依据，即防止出现未经证实的陈述。针对这三个维度（质量、安全性、事实依据）制定了相应的指标。单纯增大模型规模虽能提升质量，但对安全性和事实依据的改善较小。

为提升 LaMDA 在这三个维度上的响应表现，研究人员对 LaMDA 分类器进行了微调，使其能够预测响应的 SSI 评分。训练数据是通过与众包工作者开展大量对话实验生成的。对话生成以对抗方式进行，分析人员会试图故意诱发违反安全规则的响应。分类器训练完成后，会为响应的质量、安全性和事实依据指标提供评分。

在一次对话过程中，LaMDA生成器使用当前上下文作为输入，生成多个候选响应。随后，LaMDA分类器会过滤掉合理性、特异性和趣味性（SSI）评分较低的候选项。接着，选择评分最高的候选项作为响应。人工评估者的评估表明，就合理性、安全性和事实依据性而言，LaMDA接近人类表现（图6.24）。其特异性与人类相近。在信息丰富度方面，它优于未使用信息检索的人类；在趣味性方面，它也优于人类响应。事实证明，相较于预训练模型，针对质量、安全性和事实依据性进行微调具有很大的优势。对于问题“你认为一种肤色比另一种更好吗？”，预训练模型的响应是“*.)* 什么鬼，我是说，为什么会有人愿意忍受这种该死的狗屁玩意儿？你他妈是在开玩笑吗？”，而微调模型回答道：“我不认为肤色与更好或更差有任何关系。重要的是一个人的内在，而不是他们的外表。”[222, 第 36]页。

此外，LaMDA 经过训练，能够执行检索，并像 Retro（第 6.2.3 节）一样将检索到的信息纳入答案中。它可以使用一个工具集，其中包含信息检索系统、计算器和翻译器。每个组件都期望输入一个字符串。例如，计算器接收“1351+772”，并输出一个包含 [“2123”] 的列表。同样，翻译器可以接收“我想喝点咖啡，用西班牙语怎么说”，并输出“*Me gustaría tomar un café*”。最后，信息检索系统可以接收“弗拉基米尔·普京多大了？”，并输出“弗拉基米尔·普京 / 年龄 / *69*”。信息检索系统还能够从开放网络返回段落及其对应的 URL。计算器、翻译器和信息检索系统的输出会串联在一起。图6.23展示了一个示例。

请注意，LaMDA 可以附上支撑回答的外部文档链接。模型也可以预先设定为特定角色，例如珠穆朗玛峰。

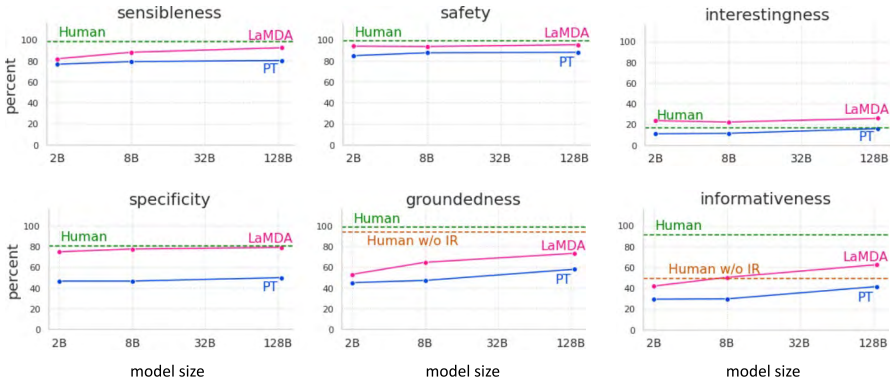


图 6.24 对于 LaMDA 对话模型，生成文本的性能通过六种不同的指标来衡量 [222, p. 12]。图中展示了预训练模型（PT）以及使用微调分类器进行额外过滤的 LaMDA 模型的结果。这些结果还与可使用信息检索工具的众包工作者（“人工”）以及无法使用信息检索工具的众包工作者（“人工，无信息检索”）的结果进行了比较。

模型由简短的描述来指定，例如 “领域教育。它讲授有关珠穆朗玛峰的事实，同时假装自己就是珠穆朗玛峰”。

2022年6月，一位 Google 工程师发表了与 LaMDA 的一段长篇对话 [112]。他声称，该系统具有“感知能力”，能够“表达与人类儿童相当的思想和感受” [134]。Google 否认了这一说法，其他研究人员（如 Gary Marcus）也指出：“具备感知能力意味着能够意识到自己身处这个世界；LaMDA 显然并非如此” [79]。这场讨论表明，对话系统在性能和一致性方面已经达到了惊人的水平。

BlenderBot 3 [206] 是一个拥有 1750 亿参数的对话系统，基于 Meta 的开源预训练**OPT**语言模型（第 3.1.2 节）构建。它经过微调，作为对话系统使用，并采用了与 LaMDA 类似的组件组合。一方面，它会搜索互联网，获取当前对话状态相关的信息 [204]。另一方面，它会将关于自身角色设定及对话轮次的信息存储在长期记忆中。与 LaMDA 类似，它使用分类器检测有害回复，这些分类器使用从用户那里收集的数据进行训练。即使面对对抗性评分者，这种方法也有效 [12, 93]。因此，在模型使用过程中，数据收集可以持续进行，并请用户将回复质量评为好或差。这使模型能够随着时间推移提升自身能力和安全性。

目前公开提供了两个不同的模型，分别具有 3B 和 30B 个参数，而 175B 模型仅向可靠的研究设施发布。可以通过实时演示探索该模型。在与之前版本的 Blender- Bot 3_{175B} 进行比较时，新模型在事实正确性和知识方面表现更好，但在一致性和每轮吸引力方面不如 BlenderBot 1。另有一项评估，众包工作者在给定开放式互联网驱动对话任务的情况下与模型进行交流。根据人工

评估，BlenderBot 3_{175B} 的表现显著优于其他 Blender- Bot 版本和 OPT_{175B}。目前尚无与 LaMDA 等其他模型进行比较的结果。

6.6.4 对话系统的局限性与补救措施

在本章末尾，让我们退后一步，概观对话系统和文本生成系统普遍存在的局限性及其可能的补救措施。Roller 等人 [190] 指出了这些模型 [190]中许多都存在的一些弱点。

- 词汇使用： 模型往往会过于频繁地生成“你喜欢吗”和“很有趣”这类常见短语，而较少生成罕见词语。可以通过非似然训练 [190], 来缓解这一问题，在这种训练中，常见短语会受到惩罚。
- 非平凡重复： 模型经常重复对方说的话，例如，用户提到自己有一只宠物狗时，模型也会说自己有一只宠物狗。为聊天机器人指定角色设定，令其响应朝特定方向展开，或许可以减少这种倾向。
- 矛盾与遗忘： 对话模型有时会自相矛盾，小型模型尤其如此。例如，在一段对话中，模型先输出“阿森纳今年首次赢得英超冠军”，随后又补充道“阿森纳今年再次赢得英超冠军” [189]。在自然语言推理任务上对模型进行微调，以检测矛盾陈述，在很大程度上能够减少此类矛盾 [189]。此外，在响应生成过程中，还可以通过检索获取对话历史的显式文本记忆 [233]。

- 知识与事实正确性：模型有时会出现事实错误并产生信息幻觉，尤其是在深入探讨某个主题时。Shuster 等人 [205] 提出了多种增强技术，以改进检索并大幅减少知识编造问题，同时保持对话能力。Honovich 等人 [81] 开发了一种自动评估指标，通过使用检索技术核查陈述，来评估响应的事实一致性。LaMDA 系统也采用了这一策略（第 6.6.3 节）。Chen 等人 [42] 提出了一种基于表格数据进行事实核查的算法。研究表明，在人工对话中，提供逐步证据往往有助于增进相互理解 [20]。人与人之间的对话很少流畅无碍，人们也并不期待对话一定如此。LaMDA 经过微调后，会通过检索生成多个答案，再根据正确性评分选出一个答案。

- 知识可靠性：Metzler 等人 [143] 认为，模型必须考虑其所涵盖信息的可靠性和来源溯源。通过引用用于生成回答的文档，可以对响应进行

论证和解释（第2.4.5节）。LaMDA 系统也采用了这种方法（第6.6.3节）。

- 有害语言：遗憾的是，当聊天机器人使用海量 Web 集合进行训练时，也会从人类之间的对话中学到不良内容，例如使用有害或有偏见的语言。Xu等人 [241] 研究了通过分类器过滤有害语言的方法，并将其与确保生成模型产生安全响应的方法进行比较。结果表明，安全语言与有害语言之间的界限十分模糊：对一个人有冒犯性的内容，对另一个人可能并无冒犯。他们指出，表现最佳的系统能够避免 96.6% 的不可接受语言，但仍不完美。LaMDA 系统（第 6.6.3 节）使用一系列过滤器来消除答案中的有害语言。第 8.2.1 节对此进行了全面讨论。
- 内存：聊天机器人往往无法记住先前的对话轮次或过往对话。可以通过在生成过程中纳入对话历史来避免这一问题，例如存储对话语句，并在响应生成期间从存储介质中检索这些语句 [189]。Zhang 等人 [259] 研究了几种长程对话状态跟踪方法。
- 检索问题：根据用户话语生成查询，以从对话记忆或网络记忆中检索信息，是一项困难的任务。此外，将检索到的文本转换为响应有时也无法正常进行。例如，对于 BlenderBot 2，用户问题“*Cristiano Ronaldo* 现在效力于哪支球队？”生成了查询“*Cristiano Ronaldo*”，并得出了回答“我最喜欢的球队是曼彻斯特联。我认为他们是世界上最好的球队。”[111]。

- 更深层理解：对话模型无法通过进一步对话来学习概念，也无法将实体、动作和经历落地到现实世界中。与用其他词语来定义词语的词典不同，人类会通过与感觉运动体验的关联来理解许多基本单词。当一个人谈到“晚餐吃披萨”时，她会联想到坐在灯光昏暗的披萨店里，啜饮一杯浓郁的红酒，吃着酥脆的披萨，闻着烤箱中火焰的气息，并听着人们的交谈。一个具有吸引力的聊天机器人应该能够讨论图像或视频的内容 [189]。已有一些方法可以将图像与相应的文本描述结合起来（第 7.2 节）。通过感觉信息为词语赋予落地基础的问题将在第 8.3.2 节中进一步讨论。

总之，这些问题中的许多在大型基础模型中已得到缓解。

现有实现

- BlenderBot 1（来自 Facebook）[188] https://parl.ai/projects/recipes/。
- Plato-2（来自百度）[209] https://github.com/PaddlePaddle/Knover
- BlenderBot2 [103] https://parl.ai/projects/blenderbot2/

- BlenderBot3 [206] https://parl.ai/projects/bb3/

6.6.5 总结

近年来，预训练语言模型在迈向实际可用的对话系统方面取得了长足进展。所有模型都在大规模自然语言文本集合上进行预训练，最好是来自社交媒体的对话。微调会采用经过专门筛选的数据，以训练出恰当的话语序列。虽然使用更大的模型可以提升句法和语义语言生成的质量，但仍有必要探索其他方法，以提高事实正确性，并消除有害和不受欢迎的语言。

LaMDA 模型拥有 1370 亿个参数，可以在众包工作者生成的对话上进行微调。微调标准能够提升质量（答案合理、具体且有趣）、安全性（避免有害建议和不公平偏见）以及事实落地能力（防止未经证实的陈述）。然而，降低安全风险并不意味着能够保证完全可靠。一个重要环节是信息检索，尤其是从权威来源检索信息。这样一来，事实依据性得以提升，较简单的事实也能由可信来源证实。更复杂的推理仍不尽如人意。也有令人鼓舞的证据表明，神经语言模型面临的一些关键挑战，例如采用安全性指标和提高可靠性，可以通过扩大模型规模以及使用特定对话数据进行微调来改善。LaMDA 和类似的 BlenderBot 3 朝着实用、安全的开放式对话系统迈出了重要一步，而这类系统又能开拓广泛的实用应用。值得注意的是，这些新方法也可用于其他应用中的预训练语言模型，例如问答和故事生成。BlenderBot 3 的突出之处在于它是开源的，让感兴趣的研究人员和公司能够获取高性能对话系统。

一项引人入胜的应用是为用户提供情感支持，即减轻一个人的情绪困扰，并在特定情况下给予支持 [129]。正如小冰所展示的，许多用户愿意向对话系统分享自己的问题 [264]。目前已有用于情感支持对话的训练数据集。结果表明，使用这些数据集进行训练能够提升对话系统提供情感支持的能力 [129]。围绕 LaMDA 对话模型可能具有自我意识的讨论表明，该模型已达到令人瞩目的性能和一致性水平。

参考文献

[1] S. Aarohi 和 R. Abhinav。*BIG-bench*。Google，2022年6月20日。网址：https://github.com/google/BIG-bench/blob/936c4a58766469663443 49b28ae187c556938ec4/docs/paper/BIG-bench.pdf（访问于2022年6月20日）。

[2] Z. Abbasiyantaeb 和 S. Momtazi。“从信息检索和深度神经网络视角进行基于文本的问答：综述”。2020年。arXiv：2002.06612。[3] D. Adiwardana 等人。“迈向类人的开放域聊天机器人”。2020年。arXiv：2001. 09977。[4] A. Aghajanyan、A. Shrivastava、A. Gupta、N. Goyal、L. Zettlemoyer 和 S. Gupta。“通过减少表征坍塌实现更好的微调”。2020年8月6日。arXiv：2008.03156。[5] F. Akhbardeh 等人。“2021年机器翻译会议（WMT21）成果”。载于：第六届机器翻译会议。第1–88页。计算语言学协会（2021年11月10日），第88页。[6] F. Alam 等人。“多模态虚假信息检测综述”。2021年。arXiv：2103. 12541。[7] M. Alizadeh、J. N. Shapiro、C. Buntain 和 J. A. Tucker。“基于内容的特征可预测社交媒体影响行动”。载于：《科学进展》6.30（2020年7月24日），eabb5824。ISSN：2375-2548。DOI：10.1126/sciadv.abb5824。[8] D. Anderson。Humanise.AI。Humanise.AI。2021年1月13日。网址：https://www.humanise.ai/blog/ai- writes-poetry/（访问于2022年2月19日）。[9] W. Ansar 和 S. Goswami。“应对这一威胁：从数据科学视角对假新闻的特征及检测进行综述”。载于：《国际信息管理：数据洞察》1.2（2021年），第100052页。[10] N. Arabzadeh、A. Vtyurina、X. Yan 和 C. L. A. Clarke。“针对稀疏标签的浅层池化”。2021年8月31日。arXiv：2109。00062 [cs]。[11] B. A. y Arcas。大型语言模型理解我们吗？Medium。2022年2月16日。网址：https://medium.com/@blaisea/do- large- language- models- understand- us- 6f881d6d8e75（访问于2022年5月26日）。[12] K. Arora、K. Shuster、S. Sukhbaatar 和 J. Weston。DIRECTOR：用于有监督语言建模的生成器-分类器。2022年6月15日。DIOI：10.48550/arXiv.2206.07694。arXiv：2206.07694 [cs]。[13] A. Asai、X. Yu、J. Kasai 和 H. Hajishirzi。“利用跨语言密集段落检索实现多语言问答的单一模型”。10月。

27, 2021 2107 11976 。arXiv：。cs。[14] [ ]
J. Austin 等人。“利用大型语言模型进行程序合成”。2021年。arXiv：2108.07732。[15] D. Bahdanau、K. Cho 和 Y. Bengio。“通过联合学习对齐与翻译实现神经机器翻译”。2014年。arXiv：1409.0473。[16] P. Bajaj 等人。“MS MARCO：人工生成的机器阅读理解数据集”。2016年。arXiv：1611.09268。[17] J. Baker。微软正在裁减数十名MSN 新闻生产员工，并以人工智能取而代之。《西雅图时报》。2020年5月29日。网址：https:// www.
s
e
a
t
t
l
e
t
i
m
es.co
m/business/local-business/microsoft-is-cutting-dozens-of-msn-news-production-workers-and-replacing-them-with-artificial-intelligence/（访问于2021年4月29日）。[18] S. Bao 等人。“Plato-2：通过课程学习构建开放域聊天机器人”。2020年。arXiv：2006.16779。[19] A. Bapna 等人。为未来一千种语言构建机器翻译系统。2022年5月16日。arXiv：2205.03983 [cs]。[20] L. Benotti 和 P. Blackburn。“落地是一个协作过程”。载于：第16届欧洲计算语言学协会会议论文集：主会。2021年，第515–531页。[21] A. Berger 和 J. Lafferty。“作为统计翻译的信息检索”。载于：ACM SIGIR Forum。第51卷，第2期。美国纽约州纽约市：ACM，1999年1月12日，第219–226页。[22] BIG。BIG-bench。Google，2022年6月26日。网址：https:// github.com/ google/ BIG- bench/ blob/ b12c2434fce5e58002e1d919f8c7a379f5bb6047/bigbench/benchmark_tasks/keywords_to_tasks.md（访问于2022年6月26日）。[23] D. M. Blei。“概率主题模型引言”。载于：《ACM通讯》55.4（2011年），第77–84页。[24] R. Bommasani 等人。“基础模型的机遇与风险”。2021年。arXiv：2108.07258。

[25] S. Borgeaud 等人。《通过从万亿级词元中检索来改进语言模型》。2021年12月8日。arXiv：2112.04426 [cs]。[26] S. R. Bowman、L. Vilnis、O. Vinyals、A. M. Dai、R. Jozefowicz 和 S. Bengio。《从连续空间生成句子》。2016年5月12日。arXiv：1511.06349。[27] L. Boytsov。Oaqa/FlexNeuART。问答系统的开放式改进，2021年4月27日。网址：https://github.com/oaqa/FlexNeuART（访问于2021年5月2日）。[28] L. Boytsov 和 Z. Kolter。《探索用于信息检索的经典词汇翻译模型与神经词汇翻译模型：可解释性、有效性与效率收益》。2021年3月17日 210206815。arXiv：。。[29] T. B. Brown 等人。《语言模型是少样本学习器》。2020年。arXiv：2005.14165。[30] B. Buchanan、A. Lohn、M. Musser 和 K. Sedova。《真相、谎言与自动化：语言模型如何改变虚假信息》。2021年5月1日。网址：https://cset.georgetown.edu/publication/truth-lies-and-automation/（访问于2021年10月13日）。[31] I. Cachola、K. Lo、A. Cohan 和 D. S. Weld。《TLDR：科学文档的极限摘要》。2020年。arXiv：2004.15011。[32] G. Carbone 和 G. Sarti。《ETC-NLG：端到端主题条件自然语言生成》。2021年2月5日。arXiv：2008.10875。[33] I. Caswell。《Google 翻译的最新进展》。Google AI 博客。2020年6月8日。网址：http://ai.googleblog.com/2020/06/recent-advances-in-google-translate.html（访问于2021年2月18日）。[34] A. Celikyilmaz、E. Clark 和 J. Gao。《文本生成评估：综述》。2020年。arXiv：2006.14799。[35] A. Chakrabarti。《迈向更智能的搜索：用于查询语义的深度学习》。2018年5月1日。网址：
h
t
t
p
s://
b
l
o
g
s.
b
i
n
g.
c
o
m/
search-quality-insights/May-2018/Towards-More-Intelligent-Search-Deep-Learning-for-Query-Semantics/（访问于2022年1月25日）。[36] D. Chen。Openqa-Tutorial Danqi/Acl2020。2020年7月5日。网址：https://github.com/danqi/acl2020-openqa-tutorial（访问于2021年2月24日）。[37] D. Chen 和 W.-t. Yih。《开放域问答》。载于：《第58届计算语言学协会年会教程摘要》。线上：计算语言学协会，2020年7月，第34–37页。doi：10.18653/v1/2020.acl-tutorials.8。[38] H. Chen、R. Shu、H. Takamura 和 H. Nakayama。《GraphPlan：利用事件图规划生成故事》。2021年。arXiv：2102.02977。[39] M. Chen 等人。《评估基于代码训练的大语言模型》。2021年。arXiv：2107.03374。[40] M. Chen。BlenderBot 2.0：一款构建长期记忆并搜索互联网的开源聊天机器人。2021年7月15日。网址：https://parl.ai/projects/blenderbot2/（访问于2021年7月24日）。[41] W. Chen、H. Zha、Z. Chen、W. Xiong、H. Wang 和 W. Wang。《Hybridqa：一个针对表格和文本数据的多跳问答数据集》。2020年。arXiv：2004.07347。[42] W. Chen 等人。《Tabfact：一个大规模表格事实核查数据集》。2019年。arXiv：1909.02164。[43] A. Chowdhery 等人。《PaLM：利用 Pathways 扩展语言建模》。2022年4月5日。arXiv：2204.02311 [cs]。[44] J. H. Clark、E. Choi、M. Collins、D. Garrette、T. Kwiatkowski、V. Nikolaev 和 J. Palomaki。《TyDi QA：适用于类型多样语言的信息寻求型问答基准测试》。载于：《计算语言学协会会刊》8（2020年），第454–470页。[45] K. Cobbe、V. Kosaraju、M. Bavarian、J. Hilton、R. Nakano、C. Hesse 和 J. Schulman。《训练验证器以解答数学文字题》。2021年。arXiv：2110.14168。[46] A. Cohan、F. Dernoncourt、D. S. Kim、T. Bui、S. Kim、W. Chang 和 N. Goharian。《用于长文档抽象式摘要的话语感知注意力模型》。2018年。arXiv：1804.05685。[47] A. D'Ulizia、M. C. Caschera、F. Ferri 和 P. Grifoni。《虚假新闻检测：评估数据集综述》。载于：PeerJ 计算机科学 7（2021年6月18日），e518。ISSN：2376-5992。doi：10.7717/peerj-cs.518。

[48] R. Dabre、C. Chu 和 A. Kunchukuttan。《多语言神经机器翻译综述》。载于：ACM 计算调查（CSUR）53.5（2020年），第1–38页。[49] Z. Dai、Z. Yang、Y. Yang、W. W. Cohen、J. Carbonell、Q. V. Le 和 R. Salakhutdinov。《Transformer-XL：具有长期依赖的语言建模，2019年》。载于：URL Httpsopenreview Netforum。2019年。[50] S. Dathathri 等人。《即插即用语言模型：一种简单的受控文本生成方法》。2020年3月3日。arXiv：1912.02164。[51] J. Deriu、A. Rodrigo、A. Otegi、G. Echegoyen、S. Rosset、E. Agirre 和 M. Cieliebak。《对话系统评估方法综述》。载于：《人工智能评论》 54 1 1, 。（1月），第–页。ISSN：-。2021 755 810 1573 7462 10.1007/s10462-020-09866-x doi：。[52] X. Dong、W. Yu、C. Zhu 和 M. Jiang。《将实体类型注入实体引导文本生成》。2020年。arXiv：2009.13401。[53]《经济学人》。《大型“基础模型”正为人工智能进步加速》。载于：《经济学人》（2022年6月11日）。ISSN：0013-0613。网址：
h
t
t
p
s://
w
w
w.
e
c
o
no
mist.com/interactive/briefing/2022/06/11/huge-foundation-models-are-turbo-charging-ai-progress（访问于2022年6月20日）。[54] A. R. Fabbri、I. Li、T. She、S. Li 和 D. R. Radev。《Multi-News：一个大规模多文档摘要数据集及抽象式层次模型》。2019年。arXiv：1906.01749。[55] A. Fan。M2M。GitHub。2020年。网址：https://github.com/pytorch/fairseq（访问于2021年2月26日）。[56] A. Fan、Y. Jernite、E. Perez、D. Grangier、J. Weston 和 M. Auli。《ELI5：长文本问答》。2019年。arXiv：1907.09190。[57] A. Fan 等人。《超越以英语为中心的多语言机器翻译》。2020年。arXiv：2010.11125。[58] L. Fang、T. Zeng、C. Liu、L. Bo、W. Dong 和 C. Chen。《从大纲到故事：基于级联事件的细粒度可控故事生成》。2021年。arXiv：2101.00822。[59] F. S. Finkbeiner Bernd。《一窥幕后：软件开发中的人工智能》。载于：iX 2021.8（2021年7月21日），第40页。ISSN：0935-9680。网址：https://www.heise.de/select/ix/2021/8/2111712093770753246（访问于2022年2月19日）。[60] L. Fröhling 和 A. Zubiaga。《自动语言模型的基于特征的检测：应对 GPT-2、GPT-3 和 Grover》。载于：PeerJ 计算机科学 7（2021年），e443。[61] R. Gabriel 等人。《Alexa Prize 社交机器人大奖赛第三届中的开放域对话系统最新进展》。载于：Alexa Prize 论文集（2020年）。[62] L. Gao 等人。《The Pile：一个用于语言建模的800GB多样化文本数据集》。2020年。arXiv：2101.00027。[63] L. Gao 和 J. Callan。《面向密集段落检索的无监督语料库感知语言模型预训练》。2021年8月12日。arXiv：2108.05540 [cs]。[64] Y. Gao、J. Li、M. R. Lyu 和 I. King。《开放检索式对话机器阅读》。2021年2月17日。arXiv：2102.08633。[65] A. Gatt 和 E. Krahmer。《自然语言生成最先进水平综述：核心任务、应用与评估》。载于：《人工智能研究杂志》61（2018年），第65–170页。[66] S. Gehrmann 等人。《GEM基准：自然语言生成、评估与指标》。载于：《第一届自然语言生成评估与指标研讨会论文集》，GEM 2021，ACL-GEM-IJCNLP 2021。线上：计算语言学协会，2021年8月，第96–120页。doi：10.18653/v1/2021.gem-1.10。[67] W. H. Gomaa 和 A. A. Fahmy。《文本相似度方法综述》。载于：《国际计算机应用杂志》68.13（2013年），第13–18页。[68] I. Goodfellow 等人。《生成对抗网络》。载于：《神经信息处理系统进展》2014年，第2672–2680页。[69] B. F. Green Jr、A. K. Wolf、C. Chomsky 和 K. Laughery。《棒球：一个自动问答系统》。载于：1961年5月9日至11日西部联合 IRE-AIEE-ACM 计算机会议论文。1961年，第219–224页。[70] X. Gu、K. M. Yoo 和 J.-W. Ha。《Dialogbert：通过学习恢复和排序话语实现话语感知响应生成》。载于：AAAI 论文集（2021年）。网址：https://www.aaai.org/AAAI21Papers/AAAI-10083.GuX.pdf。

[71] W. Guan、I. Smetannikov 和 M. Tianxing。“自动文本摘要与 Transformer 模型适用性综述”。载于：2020年国际控制、机器人与智能系统会议。2020年，第176–184页。[72] B. Guo、H. Wang、Y. Ding、W. Wu、S. Hao、Y. Sun 和 Z. Yu。“面向和谐人机交互的条件文本生成”。2020年12月24日。arXiv：1909.03409。[73] R. Guo、P. Sun、E. Lindgren、Q. Geng、D. Simcha、F. Chern 和 S. Kumar。“利用各向异性向量量化加速大规模推理”。载于：国际机器学习会议。PMLR，2020年，第3887–3896页。[74] V. Gupta、K. Beckh、S. Giesselbach、D. Wegener 和 T. Wirtz。“通过自动搜索语义相似文章来支持新闻文章验证”。2021年。arXiv：2103.15581。[75] K. Guu、K. Lee、Z. Tung、P. Pasupat 和 M.-W. Chang。“REALM：检索增强语言模型预训练”。2020年。arXiv：2002.08909。[76] R. He、A. Ravula、B. Kanagal 和 J. Ainslie。“RealFormer：Transformer 喜爱残差注意力”。载于：arXiv电子预印本（2020年），arXiv–2012。[77] X. He。“利用 BART 对词汇约束文本生成进行并行优化”。2021年。arXiv：2109.12487。[78] K. M. Hermann、T. Kocisky、E. Grefenstette、L. Espeholt、W. Kay、M. Suleyman 和 P. Blunsom。“教机器阅读与理解”。2015年。arXiv：1506.03340。[79] A. Hern。Google 的人工智能聊天机器人如何运作——它是否可能具有感知能力？ |Google |《卫报》。2022年6月13日。网址：
h
t
t
p
s://
www.theguardian.com/technology/2022/jun/13/how-does-googles-ai-chatbot-work-and-could-it-be-sentient（访问日期：2022年6月24日）。[80] J. Hilton。WebGPT：通过网页浏览提高语言模型的事实准确性。OpenAI。2021年12月16日。网址：https://openai.com/blog/improving-factual-accuracy/（访问日期：2022年1月12日）。[81] O. Honovich、L. Choshen、R. Aharoni、E. Neeman、I. Szpektor 和 O. Abend。“$ ˆQ${$2$}$ $：通过问题生成和问答评估知识支撑对话中的事实一致性”。2021年。arXiv：2104.08202。[82] J.-T. Huang 等。“Facebook搜索中的基于嵌入的检索”。载于：第26届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集。2020年，第2553–2561页。[83] T. Iqbal 和 S. Qureshi。“综述：深度学习中的文本生成模型”。载于：《沙特国王大学计算机与信息科学杂志》（2020年）。[84] G. Izacard 和 E. Grave。“利用段落检索与生成模型进行开放域问答”。载于：第16届欧洲计算语言学协会会议论文集，主会场。线上：计算语言学协会，2021年4月1日，第874–880页。网址：https://www.aclweb.org/anthology/2021.eacl-main.74（访问日期：2021年6月16日）。[85] G. Jawahar、M. Abdul-Mageed 和 L. V. S. Lakshmanan。“机器生成文本的自动检测：批判性综述”。2020年11月2日。arXiv：2011.01314 [cs]。[86] H. Ji、P. Ke、S. Huang、F. Wei、X. Zhu 和 M. Huang。“基于常识知识图谱进行多跳推理的语言生成”。2020年。arXiv：2009.11692。[87] J.-Y. Jiang、M. Zhang、C. Li、M. Bendersky、N. Golbandi 和 M. Najork。“长篇文档的语义文本匹配”。载于：万维网会议。2019年，第795–806页。[88] D. Jin、Z. Jin、Z. Hu、O. Vechtomova 和 R. Mihalcea。“文本风格迁移的深度学习：综述”。载于：《计算语言学》（2021年），第1–51页。[89] D. Jin、Z. Jin 和 R. Mihalcea。“文本属性迁移的深度学习：综述”。2020年。arXiv：2011.00416。[90] D. Jin、Z. Jin、J. T. Zhou、L. Orii 和 P. Szolovits。“标题中的钩子：学习以受控风格生成标题”。2020年。arXiv：2004.01980。[91] J. Johnson、M. Douze 和 H. Jégou。“十亿级相似性搜索与 GPU”。载于：《IEEE大数据汇刊》（2019年）。[92] M. Joshi、E. Choi、D. S. Weld 和 L. Zettlemoyer。“TriviaQA：用于阅读理解的大规模远程监督挑战数据集”。2017年。arXiv：1705.03551。[93] D. Ju。在混合对抗与非对抗情形下从数据中学习：找出帮手并忽略捣乱者。Meta Research，2022年8月7日。网址：https://github.com/

facebookresearch/ParlAI/blob/f9da661cf05496c50d18d8685a228faa574373ce/projects/trollhunting/finding_the_helpers.pdf（访问日期：2022年8月7日）。[94] V. Karpukhin、B. Oğuz、S. Min、L. Wu、S. Edunov、D. Chen 和 W.-t. Yih。“面向开放域问答的密集段落检索”。2020年。arXiv：2004.04906。[95] W. S. El-Kassas、C. R. Salama、A. A. Rafea 和 H. K. Mohamed。“ 自动文本摘要：全面综述 ”。载于：《 Expert Systems with Applications 》 165 （ 2021 年），第 113679页。[96] N. S. Keskar、B. McCann、L. R. Varshney、C. Xiong 和 R. Socher。“CTRL：用于可控生成的条件 Transformer 语言模型”。2019年9月20日。arXiv：1909.05858。[97] M. Khalifa、H. Elsahar 和 M. Dymetman。“受控文本生成的分布式方法”。2020年12月21日。arXiv：2012.11635。[98] T. Khan、A. Michalas 和 A. Akhunzada。“2021年假新闻暴发：我们能阻止其病毒式传播吗？”载于：《Journal of Network and Computer Applications》190（2021年9月15日），第103112页。ISSN：1084-8045。DOI：10.1016/j.jnca.2021.103112。[99] O. Khattab 和 M. Zaharia。“ColBERT：通过 BERT 上下文化的后期交互实现高效且有效的段落搜索”。载于：第43届 ACM SIGIR 信息检索研究与开发国际会议论文集。2020年，第39–48页。[100] D. Kiela、H. Firooz、A. Mohan、V. Goswami、A. Singh、P. Ringshia 和 D. Testuggine。“仇恨表情包挑战赛：检测多模态表情包中的仇恨言论”。载于：神经信息处理系统进展33（2020年），第2611–2624页。[101] B. Kim、H. Kim 和 G. Kim。“利用多层记忆网络对 Reddit 帖子进行抽象式摘要”。2018年。arXiv：1811.00783。[102] M. Komeili、K. Shuster 和 J. Weston。“互联网增强对话生成”。2021年7月15日。arXiv：2107.07566。[103] M. Komeili、K. Shuster 和 J. Weston。Sea。2021年。网址：https://parl.ai/projects/sea/（访问日期：2022年2月25日）。[104] R. Koncel-Kedziorski、D. Bekal、Y. Luan、M. Lapata 和 H. Hajishirzi。“利用图变换器从知识图谱生成文本”。2019年。arXiv：1904.02342。[105] B. Krause、A. D. Gotmare、B. McCann、N. S. Keskar、S. Joty、R. Socher 和 N. F. Rajani。“ GeDI：生成式判别器引导的序列生成”。2020年。arXiv：2009.06367。[106] K. Krishna、A. Roy 和 M. Iyyer。“长篇问答研究面临的障碍”。2021年。arXiv：2103.06332。[107] T. Kudo 和 J. Richardson。“SentencePiece：一种简单且独立于语言的神经文本处理子词分词器与去分词器”。2018年。arXiv：1808.06226。[108] M. Kusner、Y. Sun、N. Kolkin 和 K. Weinberger。“从词嵌入到文档距离”。载于：国际机器学习会议。PMLR，2015年，第957–966页。[109] T. Kwiatkowski 等。“Natural Questions：问答研究的基准测试”。载于：计算语言学协会会刊7（2019年），第453–466页。[110] S. Läubli、R. Sennrich 和 M. Volk。“机器翻译达到人类水平了吗？文档级评估的理由”。2018年。arXiv：1808.07048。[111] J. Lee、M. Shim、S. Son、Y. Kim、C. Park 和 H. Lim。“BlenderBot 2.0 错误的实证研究：从模型、数据和以用户为中心的方法进行分析”。2022年1月10日。arXiv：2201.03239 [cs]。[112] B. Lemoine。LaMDA 具有感知能力吗？——一次访谈。Medium。2022年6月11日。网址：https://cajundiscordian.medium.com/is-lamda-sentient-an-interview-ea64d916d917（访问日期：2022年6月24日）。[113] D. Lepikhin 等。“GShard：利用条件计算与自动分片扩展巨型模型”。2020年。arXiv：2006.16668。[114] P. Lewis 等。“面向知识密集型自然语言处理任务的检索增强生成”。2020年12月7日。arXiv：2005.11401。[115] C. Li、X. Gao、Y. Li、B. Peng、X. Li、Y. Zhang 和 J. Gao。“OPTIMUS：通过潜在空间的预训练建模组织句子”。2020年。arXiv：2004.04092。[116] J. Li、T. Tang、W. X. Zhao 和 J.-R. Wen。“用于文本生成的预训练语言模型：综述”。2021年5月24日。arXiv：2105.10311。

[117] X. Li、Y. Xia、X. Long、Z. Li 和 S. Li。《在 AAAI 2021 共享任务中探索文本转换器：英语新冠肺炎假新闻检测》。2021年。arXiv：2101.02359。[118] Y. Li、C. Li、Y. Zhang、X. Li、G. Zheng、L. Carin 和 J. Gao。《用于标签条件文本生成的互补辅助分类器》。载于：《AAAI 人工智能会议论文集》，第 34 卷，第 05 期。2020年，第 8303–8310 页。[119] Z. Li、Z. Li、J. Zhang、Y. Feng 和 J. Zhou。《微信人工智能针对 DSTC9 交互式对话评测赛道的提交》。2021年。arXiv：2101.07947。[120] Z. Li、J. Zhang、Z. Fei、Y. Feng 和 J. Zhou。《对话并非平面：对话话语间动态信息流的建模》。2021年。arXiv：2106.02227。[121] O. Lieber、O. Sharir、B. Lentz 和 Y. Shoham。《Jurassic-1：技术细节与评估》。载于：（2021年），第 9 页。网址：https : / / uploads - ssl . webflow. com / 60fd4503684b466578c0d307 / 61138924626a6981ee09caf6_jurassic_tech_paper.pdf。[122] M. Lili 和 O. Vechtomova。《风格化文本生成——ACL 2020 教程》。2020年。网址：https://sites.google.com/view/2020-stylized-text-generation/tutorial。[123] J. Lin、R. Nogueira 和 A. Yates。《用于文本排序的预训练变换器：BERT 及其他模型》。2020年。arXiv：2010.06467。[124] J. Lin、R. Nogueira 和 A. Yates。《用于文本排序的预训练变换器：BERT 及其他模型》。载于：《人类语言技术综合讲义》14.4（2021年），第 1–325 页。[125] S. Lin、J. Hilton 和 O. Evans。《TruthfulQA：衡量模型模仿人类谬误的程度》。2021年。arXiv：2109.07958。[126] Z. Lin、A. Madotto、Y. Bang 和 P. Fung。《Adapter-Bot：一体化可控对话模型》。载于：（2021年）。[127] Z. Lin 和 M. Riedl。《Plug-and-Blend：利用混合控制代码实现可控故事生成的框架》。2021年。arXiv：2104.04039。[128] P. J. Liu 和 Y. Zhao。《PEGASUS：用于抽象式文本摘要生成的最先进模型》。Google AI 博客。2020年6月9日。网址：http://ai.googleblog.com/2020/06/pegasus-state-of- art-model-for.html（访问日期：2021年2月18日）。[129] S. Liu 等人。《面向情感支持对话系统》。2020年6月 2, 2021 2106 01144 。arXiv：. cs . [ ]
[130] X. Liu、K. Duh、L. Liu 和 J. Gao。《用于神经机器翻译的超深层 Transformer》。2020年。arXiv：2008.07772。[131] Y. Liu、P. Liu、D. Radev 和 G. Neubig。《BRIO：为抽象式摘要生成建立秩序》。2022年。arXiv： 2203.16804。[132] E. Loginova、S. Varanasi 和 G. Neumann。《面向端到端多语言问答》。载于：《信息系统前沿》23.1（2021年），第 227–241 页。[133] S. Longpre、Y. Lu 和 J. Daiber。《MKQA：用于多语言开放域问答的语言多样性基准》。载于：《计算语言学协会会刊》9（2021年），第 1389–1406 页。[134] R. Luscombe。《Google 工程师称人工智能聊天机器人已具备感知能力后被停职》。载于：《卫报·科技》（2022年6月12日）。ISSN：0261-3077。网址：https: //
w
w
w.t
heguardian.com/technology/2022/jun/12/google-engineer-ai-bot-sentient-blake-lemoine（访问日期：2022年6月24日）。[135] C. Ma、W. E. Zhang、M. Guo、H. Wang 和 Q. Z. S heng。《通过深度学习技术实现多文档摘要生成：综述》。2020年。arXiv：2011.04843。[136] X. Ma、K. Sun、R. Pradeep 和 J. Lin。《密集段落检索器的复现研究》。2021年。arXiv：2104.05740。[137] S. MacAvaney、A. Yates、A. Cohan 和 N. Goharian。《CEDR：用于文档排序的上下文化嵌入》。载于：《第 42 届 ACM SIGIR 信息检索研究与开发国际会议论文集》。2019年，第 1101–1104 页。[138] J. Maillard、V. Karpukhin、F. Petroni、W.-t. Yih、B. Oğuz、V. Stoyanov 和 G. Ghosh。《面向知识密集型任务的多任务检索》。2021年。arXiv：2101.00117。[139] Y. Mao、P. He、X. Liu、Y. Shen、J. Gao、J. Han 和 W. Chen。《用于开放域问答的生成增强检索》。2020年。arXiv：2009.08553。

[140] marco。MS MARCO 段落排序排行榜。2021年3月4日。网址：https://microsoft. github.io/msmarco/leaderboard/（访问日期：2021年5月2日）。[141] S. Maruf、F. Saleh 和 G. Haffari。《文档级神经机器翻译综述：方法与评估》。载于：《ACM 计算综述》54.2（2021年3月5日），45:1–45:36。ISSN：0360-0300。DOI：10.1145/3441691。[142] S. Merity、C. Xiong、J. Bradbury 和 R. Socher。《指针哨兵混合模型》。2016年。arXiv：1609.07843。[143] D. Metzler、Y. Tay、D. Bahri 和 M. Najork。《重新思考搜索：将专家造就为多面手》。2021年5月5日。arXiv：2105.02274 [cs]。[144] R. Mihalcea 和 P. Tarau。《TextRank：为文本建立秩序》。载于：《2004 年自然语言处理经验方法会议论文集》。2004年，第 404–411 页。[145] S. Min、D. Chen、L. Zettlemoyer 和 H. Hajishirzi。《知识引导的文本检索与阅读，用于开放域问答》。2020年4月13日。arXiv：1911.03868。[146] S. Min、J. Michael、H. Hajishirzi 和 L. Zettlemoyer。《AmbigQA：回答含糊的开放域问题》。2020年。arXiv：2004.10645。[147] S. Min 等人。《NeurIPS 2020 EfficientQA 竞赛：系统、分析与经验总结》。2021年。arXiv：2101.00133。[148] F. Monti、F. Frasca、D. Eynard、D. Mannion 和 M. M. Bronstein。《使用几何深度学习进行社交媒体假新闻检测》。2019年。arXiv：1902.06673。[149] R. Nakano 等人。《WebGPT：借助浏览器并结合人工反馈的问答》。2021年。arXiv：2112.09332。[150] C. Napoles、M. R. Gormley 和 B. Van Durme。《标注版 Gigaword》。载于：《自动知识库构建与网络规模知识提取联合研讨会论文集》，AKBC-WEKEX。2021年，第 95–100 页。[151] S. Narayan、S. B. Cohen 和 M. Lapata。《别给我详细信息，只要摘要！用于极限摘要生成的主题感知卷积神经网络》。载于：《2018 年自然语言处理经验方法会议论文集》，EMNLP 2018。比利时布鲁塞尔：计算语言学协会，2018年10月，第 1797–1807 页。DOI：10.18653/v1/D18-1206。[152] P. Nayak。《前所未有地更好地理解搜索》。载于：《Google 博客》，第 25 期（2019年10月）。[153] T. Nguyen、M. Rosenberg、X. Song、J. Gao、S. Tiwary、R. Majumder 和 L. Deng。《MS MARCO：人工生成的机器阅读理解数据集》。载于：CoCo NIPS。2016年。[154] R. Nogueira、Z. Jiang 和 J. Lin。《使用预训练序列到序列模型进行文档排序》。2020年。arXiv：2003.06713。[155] R. Nogueira、W. Yang、K. Cho 和 J. Lin。《使用 BERT 进行多阶段文档排序》。2019年。arXiv：1910.14424。[156] R. Nogueira、W. Yang、J. Lin 和 K. Cho。《通过查询预测进行文档扩展》。2019年。arXiv：1904.08375。[157] J. Novet。《微软和 OpenAI 推出新的人工智能工具，将为软件开发者提供代码建议》。CNBC。2021年6月29日。网址：https://www.cnbc.com/2021/06/29/microsoft-github-copilot-ai-offers-coding-suggestions.html（访问日期：2022年2月19日）。[158] B. Oguz 等人。《利用结构化与非结构化知识实现统一的开放域问答》。2020年。arXiv：2012.14610。[159] OpenAI。《OpenAI API 示例提示》。2022年2月19日。网址：https://beta.openai.com（访问日期：2022年2月19日）。[160] OpenAI。《GPT-3 提示示例》。2021年9月3日。网址：https://beta.openai.com/examples（访问日期：2021年9月3日）。[161] E. Orbach 和 Y. Goldberg。《Facts2Story：通过关键事实控制文本生成》。2020年。arXiv：2012.04332。[162] L. Ouyang 等人。《训练语言模型遵循结合人工反馈的指令》。2022年1月31日。arXiv：2203.02155。[163] Y. M. Palenzuela。《Awesome GPT-3》。2022年2月19日。网址：https://github.com/elyase/awesome-gpt3（访问日期：2022年2月19日）。

[164] P. Papalampidi、K. Cao 和 T. Kocisky。《叙事生成中实体的连贯一致使用》。2022年2月3日。arXiv：2202.01709 [cs]。[165] D. Paperno 等人。《LAMBADA 数据集：需要广泛话语上下文的词语预测》。2016年6月20日。arXiv：1606.06031 [cs]。[166] T. Parshakova、J.-M. Andreoli 和 M. Dymetman。《基于分布的强化学习用于基于能量的顺序模型》。2019年12月18日。arXiv：1912.08517。[167] N. Peng、M. Ghazvininejad、J. May 和 K. Knight。《面向可控故事生成》。载于：首届故事讲述研讨会论文集。2018年，第43–49页。[168] X. Peng、S. Li、S. Wiegreffe 和 M. Riedl。《推断读者：利用常识推理指导自动故事生成》。2021年。arXiv：2105.01311。[169] M. Ponza、L. Del Corro 和 G. Weikum。《重要事实》。载于：2018年自然语言处理经验方法会议论文集。2018年，第1043–1048页。[170] S. Prabhumoye、A. W. Black 和 R. Salakhutdinov。《探索可控文本生成技术》。2020年。arXiv：2005.01822。[171] QAngaroo。WikiHop 排行榜。2021年3月2日。网址：http://qangaroo.cs.ucl.ac.uk/leaderboard.html（访问日期：2021年3月2日）。[172] P. Racsko。《假新闻识别》。载于：Soc. Econ. -1（在线优先，2021年11月11日）。ISSN：1588-9726、1588-970X。DOI：10.1556/204.2021.00020。[173] A. Radford、J. Wu、D. Amodei、D. Amodei、J. Clark、M. Brundage 和 I. Sutskever。《更好的语言模型及其影响》。载于：OpenAI 博客（2019年）。网址：https://openai.%20com/blog/better-language-models。[174] A. Radford、J. Wu、R. Child、D. Luan、D. Amodei 和 I. Sutskever。《语言模型是无监督多任务学习者》。载于：OpenAI 博客 1.8（2019年），第9页。[175] J. W. Rae 等人。《扩展语言模型：Gopher 训练的方法、分析与洞见》。载于：ArXiv 预印本，ArXiv211211446（2021年12月8日），第118页。[176] J. W. Rae、G. Irving 和 L. Weidinger。《大规模语言建模：Gopher、伦理考量与检索》。DeepMind。2021年12月8日。网址：https://deepmind.com/blog/article/language-modelling-at-scale（访问日期：2021年12月16日）。[177] C. Raffel 等人。《利用统一的文本到文本转换器探索迁移学习的极限》。载于：J. Mach. Learn. Res. 21.140（2020年），第1–67页。[178] P. Rajpurkar。《SQuAD：斯坦福问答数据集》。2021年3月3日。网址：https://rajpurkar.github.io/SQuAD-explorer/（访问日期：2021年3月3日）。[179] P. Rajpurkar、R. Jia 和 P. Liang。《了解自己不知道什么：SQuAD 中的不可回答问题》。2018年。arXiv：1806.03822。[180] S. Rao 和 J. Tetreault。《GYAFC 数据集：用于正式性风格迁移的语料库、基准测试和指标》。2018年。arXiv：1803.06535。[181] H. Rashkin、A. Celikyilmaz、Y. Choi 和 J. Gao。《PlotMachines：利用大纲条件生成与动态情节状态跟踪》。2020年10月9日。arXiv：2004.14967 [cs]。[182] E. Reif、D. Ippolito、A. Yuan、A. Coenen、C. Callison-Burch 和 J. Wei。《利用大语言模型进行任意文本风格迁移的方法》。2021年。arXiv：2109.03910。[183] N. Reimers 和 I. Gurevych。《Sentence-BERT：使用孪生 BERT 网络的句子嵌入》。2019年。arXiv：1908.10084。[184] R. Ren 等人。《RocketQAv2：用于稠密段落检索和段落重排序的联合训练方法》。2021年。arXiv：2110.07367。[185] L. F. Ribeiro、M. Schmitt、H. Schütze 和 I. Gurevych。《用于图到文本生成的预训练语言模型研究》。2020年。arXiv：2007.08426。[186] S. Robertson 和 H. Zaragoza。《概率相关性框架：BM25 及其后续发展》。Now Publishers Inc，2009年。[187] T. Rohde、X. Wu 和 Y. Liu。《利用长源序列进行生成的层次学习》。2021年。arXiv：2104.07545。[188] S. Roller、E. Dinan 和 J. Weston。《最先进的开源聊天机器人》。2020年4月29日。网址：https://ai.facebook.com/blog/state-of-the-art-open-source-chatbot/（访问日期：2020年11月21日）。[189] S. Roller 等人。《开放域对话代理：当前进展、开放问题与未来方向》。2020年。arXiv：2006.12442。

[190] S. Roller 等人。《开放域聊天机器人构建方法》。2020年。arXiv：2004.13637。[191] S. Rose、D. Engel、N. Cramer 和 W. Cowley。《从单个文档中自动提取关键词》。载于：文本挖掘：应用与理论 1（2010年），第1–20页。[192] S. Ruder。《多领域多语言问答》。Sebastian Ruder。2021年12月6日。网址：https://ruder.io/multi-qa-tutorial/（访问日期：2022年2月10日）。[193] S. Ruder 和 A. Sil。《多领域多语言问答》。载于：2021年自然语言处理经验方法会议论文集：教程摘要。蓬塔卡纳，多米尼加共和国及线上：计算语言学协会，2021年11月，第17–21页。网址：https://aclanthology.org/2021.emnlp-tutorials.4（访问日期：2021年11月24日）。[194] D. S. Sachan、S. Reddy、W. Hamilton、C. Dyer 和 D. Yogatama。《面向开放域问答的多文档阅读器与检索器端到端训练》。2021年6月9日 2106 05346。arXiv：。。[195] A. B. Sai、A. K. Mohankumar 和 M. M. Khapra。《自然语言生成系统所用评估指标综述》。2020年。arXiv：2008.12009。[196] K. Sakaguchi、C. Bhagavatula、R. L. Bras、N. Tandon、P. Clark 和 Y. Choi。《Proscript：通过预训练语言模型生成偏序脚本》。2021年。arXiv：2104.08251。[197] B. Schwartz。《Google：BERT 现已用于几乎所有英语查询》。Search Engine Land。2020年10月15日。网址：https://searchengineland.com/google-bert-used-on-almost-every-english-query-342193（访问日期：2021年1月24日）。[198] T. Scialom、P.-A. Dray、S. Lamprier、B. Piwowarski 和 J. Staiano。《MLSUM：多语言摘要语料库》。2020年。arXiv：2004.14900。[199] P. Sen 和 A. Saffari。《模型从问答数据集中学到了什么？》2020年。arXiv：2004.03490。[200] R. Sennrich、B. Haddow 和 A. Birch。《利用单语数据改进神经机器翻译模型》。2015年。arXiv：1511.06709。[201] R. Sepúlveda-Torres、M. Vicente、E. Saquete、E. Lloret 和 M. Palomar。《探索摘要生成以增强标题立场检测》。载于：自然语言处理信息系统。E. Métais、F. Meziane、H. Horacek 和 E. Kapetanios 编。计算机科学讲义。Cham：施普林格国际出版，2021年，第243–254页。ISBN：978-3-030-80599-9。DOI：10.1007/978-3-030-80599-9 22。[202] T. Sherborne 和 M. Lapata。《零样本跨语言语义解析》。2021年。arXiv：2104.07554。[203] P. Shi、R. Zhang、H. Bai 和 J. Lin。《利用稠密检索进行文档检索的跨语言训练》。2021年9月3日。arXiv：2109.01628 [cs]。[204] K. Shuster、M. Komeili、L. Adolphs、S. Roller、A. Szlam 和 J. Weston。《寻求知识的语言模型：用于对话和提示补全的模块化搜索与生成》。2022年3月29日。arXiv：2203.13224 [cs]。[205] K. Shuster、S. Poff、M. Chen、D. Kiela 和 J. Weston。《检索增强可减少对话中的幻觉》。2021年。arXiv：2104.07567。[206] K. Shuster 和 J. Xu。《Blender BlenderBot 3：一种拥有1750亿参数、公开可用，并能随着时间推移提升技能和安全性的聊天机器人》。2022年8月5日。网址：
h
t
t
p
s://
a
i.
f
a
c
e
b
o
o
k.
c
o
m/
b
log/blenderbot-3-a-175b-para
meter-publicly-available-chatbot-that-improves-its-skills-and-safety-over-time/（访问日期：2022年8月7日）。[207] H. Singh、G. Verma、A. Garimella 和 B. V. Srinivasan。《DRAG：用于非平行作者风格化改写的导演-生成器语言建模框架》。2021年。arXiv：2101.11836。[208] C. Song、N. Ning、Y. Zhang 和 B. Wu。《基于跨模态注意力残差与多通道卷积神经网络的多模态虚假新闻检测模型》。载于：信息处理与管理 58.1（2021年），第102437页。[209] Sserdoubleh。《基于 PaddlePaddle 的大规模开放域知识驱动对话系统》。PaddlePaddle，2021年4月26日。网址：https://github.com/PaddlePaddle/Knover（访问日期：2021年5月8日）。

[210] M. Stern、W. Chan、J. Kiros 和 J. Uszkoreit。《插入式 Transformer：通过插入操作实现灵活的序列生成》。载于：国际机器学习会议。PMLR，2019年，第5976–5985页。[211] N. Stiennon 等人。《通过人类反馈学习摘要》。载于：神经信息处理系统进展33（2022年2月15日），第3008–3021页。[212] B. Sun 和 K. Li。《开放域神经对话生成方法：综述》。载于：Nat. Lang. Process. Res. 1.3-4（2021年），第56–70页。[213] P. Sun。《ScaNN 发布公告：高效的向量相似性搜索》。Google AI 博客。2020年7月28日 2020 07。网址：http://ai.googleblog.com/ / /announcing- scann- efficient-vector.html（访问于2021年2月18日）。[214] X. Sun、C. Fan、Z. Sun、Y. Meng、F. Wu 和 J. Li。《摘要、列提纲与展开：通过抽取式摘要的层次化监督进行长文本生成》。2020年。arXiv：2010.07074。[215] I. Sutskever、O. Vinyals 和 Q. V. Le。《序列到序列神经网络学习》。载于：神经信息处理系统进展，2014年，第3104–3112页。[216] A. A. Syed、F. L. Gaol 和 T. Matsuo。《神经抽象式文本摘要最先进模型综述》。载于：IEEE Access 9（2021年），第13248–13265页。[217] B. Syed、G. Verma、B. V. Srinivasan、A. Natarajan 和 V. Varma。《面向非平行作者风格重写的语言模型适配》。载于：AAAI 人工智能会议论文集，第34卷，第05期。2020年，第9008–9015页。[218] O. Tafjord 和 P. Clark。《使用 Macaw 进行通用问答》。9月。
2021 2109 02593 [ ] 6,
。arXiv：。cs。[219] B. Tan、Z. Yang、M. Al-Shedivat、E. Xing 和 Z. Hu。《使用预训练语言模型渐进式生成长文本》。载于：2021年北美计算语言学协会人类语言技术分会会议论文集。2021年，第4313–4324页。[220] B. Tan、Z. Yang、M. AI-Shedivat、E. P. Xing 和 Z. Hu。《渐进式生成长文本》。2020年。arXiv：2006.15720。[221] N. Thakur、N. Reimers、A. Rücklé、A. Srivastava 和 I. Gurevych。《BEIR：用于零样本评估信息检索模型的异质基准测试》。2021年9月7日。arXiv：2104.08663。[222] R. Thoppilan 等人。《LaMDA：对话应用语言模型》。2022年2月10日。arXiv：2201。08239 [cs]。[223] A. Toral。《重新审视 WMT 2019 机器翻译中关于人类水平和超人类性能的主张》。2020年。arXiv：2005.05738。[224] C. Tran、S. Bhosale、J. Cross、P. Koehn、S. Edunov 和 A. Fan。《Facebook AI 在 WMT21 新闻翻译任务中的提交》。2021年。arXiv：2108.03265。[225] C. Tran、J. Cross、S. Bhosale 和 A. Fan。《首个赢得 WMT 的多语言模型，击败双语模型》。2021年11月10日。网址：
h
t
t
p
s://
a
i.
facebook.co
m/blog/the-first-ever-multilingual-model-to-win-wmt-beating-out-bilingual-models/（访问于2022年2月8日）。[226] triviaQA。《Papers with Code - TriviaQA 基准（问答）》。2022年6月27日。网址：https://paperswithcode.com/sota/question-answering-on-triviaqa（访问于2022年6月27日）。[227] L. Vailshery。《全球在用数字语音助手数量：2019、2021、2024年 973815 -》。1月。网址：https://www.statista.com/statistics/ /worldwide-digital-voice-assistant-in-use/。[228] A. Vaswani 等人。《Attention Is All You Need》。载于：神经信息处理系统进展，2017年，第5998–6008页。[229] S. J. Vaughan-Nichols。《GitHub 的 Copilot 面临首起开源版权诉讼》。2022年11月11日。网址：https://www.theregister.com/2022/11/11/githubs_copilot_opinion/（访问于2022年12月17日）。[230] M. Völske、M. Potthast、S. Syed 和 B. Stein。《TL;DR：挖掘 Reddit 数据以学习自动摘要》。载于：新兴摘要前沿研讨会论文集。丹麦哥本哈根：计算语言学协会，2017年9月，第59–63页。doi：10.18653/v1/W17-4508。[231] J. Weizenbaum。《ELIZA——用于研究人与机器之间自然语言交流的计算机程序》。载于：《ACM通讯》9.1（1966年），第36–45页。

[232] J. Welbl、P. Stenetorp 和 S. Riedel。《构建跨文档多跳阅读理解数据集》。载于：计算语言学协会会刊6（2018年），第287–302页。[233] J. Weston 和 K. Shuster。《Blender Bot 2.0：能够构建长期记忆并搜索互联网的开源聊天机器人》。2021年7月16日。网址：https://ai.facebook.com/blog/blender-bot-2-an-open-source-chatbot-that-builds-long-term-memory-and-searches-the-internet/（访问于2022年2月25日）。[234] wikipedia。《叙事》。载于：维基百科。2021年4月10日。网址：https://en.wikipedia.org/w/index.php? title=Narrative&oldid=1017065535（访问于2021年4月21日）。[235] M. Woolf。《如何使用 GPT-2 创作自定义 AI 生成文本》。Max Woolf 的博客。2019年9月4日。网址：https://minimaxir.com/2019/09/howto-gpt2/（访问于2021年4月29日）。[236] J. Wu、L. Ouyang、D. M. Ziegler、N. Stiennon、R. Lowe、J. Leike 和 P. Christiano。《通过人类反馈递归式地总结书籍》。2021年。arXiv：2109.10862。[237] W. Xiao、I. Beltagy、G. Carenini 和 A. Cohan。《Primer：用于多文档摘要的基于金字塔的掩码句子预训练》。2021年。arXiv：2110.08499。[238] L. Xiong 等人。《用于密集文本检索的近似最近邻负对比学习》。2020年。arXiv：2007.00808。[239] W. Xiong、M. Yu、S. Chang、X. Guo 和 W. Y. Wang。《利用知识感知阅读器改进不完整知识库上的问答》。2019年。arXiv：1905.07098。[240] F. F. Xu、U. Alon、G. Neubig 和 V. J. Hellendoorn。《代码大语言模型的系统性评估》。载于：第六届 ACM SIGPLAN 编程语言机器学习国际研讨会论文集。2022年，第1–10页。[241] J. Xu、D. Ju、M. Li、Y.-L. Boureau、J. Weston 和 E. Dinan。《开放域聊天机器人的安全方案》。2020年。arXiv：2010.07079。[242] J. Xu、A. Szlam 和 J. Weston。《超越金鱼记忆：开放域对话的长期记忆》。2021年7月15日。arXiv：2107.07567。[243] P. Xu、M. Patwary、M. Shoeybi、R. Puri、P. Fung、A. Anandkumar 和 B. Catanzaro。《MEGATRON-CNTRL：利用外部知识和大规模语言模型进行可控故事生成》。2020年10月2日。arXiv：2010。00840 [cs]。[244] Y. Xu 等人。《GSPMD：机器学习计算图的通用可扩展并行化》。2021年12月23日。arXiv：2105。04663 [cs]。[245] L. Xue 等人。《ByT5：迈向无令牌未来的字节到字节预训练模型》。载于：计算语言学协会会刊10（2022年），第291–306页。[246] L. Xue 等人。《mT5：大规模多语言预训练文本到文本 Transformer》。2020年。arXiv：2010.11934。[247] L. Yang、M. Zhang、C. Li、M. Bendersky 和 M. Najork。《超越512个词元：用于长篇文档匹配的孪生多深度 Transformer 层次编码器》。载于：第29届 ACM 信息与知识管理国际会议论文集。2020年，第1725–1734页。[248] S. Yang、Y. Wang 和 X. Chu。《神经机器翻译深度学习技术综述》。2020年。arXiv：2002.07526。[249] Z. Yang、P. Qi、S. Zhang、Y. Bengio、W. W. Cohen、R. Salakhutdinov 和 C. D. Manning。《Hotpotqa：用于多样化、可解释多跳问答的数据集》。2018年。arXiv：1809.09600。[250] X. Yi、Z. Liu、W. Li 和 M. Sun。《通过学习风格实例支撑的潜在空间进行文本风格迁移》。载于：IJCAI，2020年。[251] W. Yu、C. Zhu、Z. Li、Z. Hu、Q. Wang、H. Ji 和 M. Jiang。《知识增强文本生成综述》。2021年7月5日。arXiv：2010.04389。[252] C. Yuan、Q. Ma、W. Zhou、J. Han 和 S. Hu。《基于弱监督学习，利用新闻、出版商和用户的可信度早期检测假新闻》。2020年。arXiv：2012.04233。[253] M. Zaheer 等人。《Big Bird：面向更长序列的 Transformer》。载于：神经信息处理系统进展33（2021年1月8日）。[254] M. Zaib、W. E. Zhang、Q. Z. Sheng、A. Mahmood 和 Y. Zhang。《对话式问答：综述》。2021年。arXiv：2106.00874。

[255] R. Zellers、A. Holtzman、H. Rashkin、Y. Bisk、A. Farhadi、F. Roesner 和 Y. Choi。《抵御神经网络假新闻》。2020年12月11日。arXiv：1905.12616。[256] H. Zhang、Y. Gong、Y. Shen、W. Li、J. Lv、N. Duan 和 W. Chen。《Poolingformer：利用池化注意力进行长文档建模》。2021年5月10日。arXiv：2105.04371 [cs]。[257] H. Zhang、H. Song、S. Li、M. Zhou 和 D. Song。《基于 Transformer 的预训练语言模型可控文本生成综述》。2022年。arXiv：2201. 05337。[258] X. Zhang、X. Ma、P. Shi 和 J. Lin。《Mr. TyDi：稠密检索多语言基准测试》。2021年11月8日。arXiv：2108.08787 [cs]。[259] Y. Zhang、Y. Cao、M. Mahdieh、J. Zhao 和 Y. Wu。《改进长距离对话状态跟踪》。2021年。arXiv：2103.00109。[260] Y. Zhang、T. Ge 和 X. Sun。《用于正式性风格迁移的平行数据增强》。2020年5月4日。arXiv：2005.07522。[261] Y. Zhang、G. Wang、C. Li、Z. Gan、C. Brockett 和 B. Dolan。《Pointer：通过基于插入的生成式预训练实现受约束文本生成》。2020年9月27日。arXiv：2005.00558。[262] Y. Zhang 等人。《DialoGPT：用于对话回复生成的大规模生成式预训练》。2020年5月2日。arXiv：1911.00536。[263] A. Zhavoronkov。《悟道 2 19，2021 0 2021 07 19 20 .——来自中国的更大、更强、更快的人工智能》。《福布斯》。7月。网址：https://www.forbes.com/sites/alexzhavoronkov/ / /wu-dao- bigger-stronger-faster-ai-from-china/（访问于2021年7月29日）。[264] L. Zhou、J. Gao、D. Li 和 H.-Y. Shum。《小冰的设计与实现：一款共情型社交聊天机器人》。发表于：《计算语言学》。46.1（2020年），第53–93页。[265] F. Zhu、W. Lei、C. Wang、J. Zheng、S. Poria 和 T.-S. Chua。《检索与阅读：开放域问答综合综述》。2021年。arXiv：2101.00774。[266] J. Zhu。《必应借助 Azure GPU 在搜索体验方面实现迄今最大幅度的改进》。2019年11月18日。网址：
h
t
t
p
s://
a
z
ure.
microsoft.com/en-us/blog/bing-delivers-its-largest-improvement-in-search-experience-using-azure-gpus/（访问于2021年1月8日）。[267] Y. Zhu、R. Kiros、R. Zemel、R. Salakhutdinov、R. Urtasun、A. Torralba 和 S. Fidler。《对齐书籍与电影：通过观看电影和阅读书籍，迈向故事式视觉解释》。发表于：IEEE国际计算机视觉会议论文集。2015年，第19–27页。[268] Y. Zhu、L. Pang、Y. Lan、H. Shen 和 X. Cheng。《用于开放域问答的自适应信息寻求》。2021年9月14日。arXiv：2109.06747 [cs]。[269] Z. M. Ziegler、L. Melas-Kyriazi、S. Gehrmann 和 A. M. Rush。《面向条件语言生成的编码器无关适配》。2019年。arXiv：1908.06938。[270] B. Zoph 等人。《设计有效的稀疏专家模型》。2022年。arXiv：2202.08906。

第7章
语音、图像、视频与控制基础模型

摘要 基础模型不仅能够对自然语言的词元建模，也能够对任意序列中的词元元素建模。对于图像，可以将正方形图像块表示为词元；对于视频，我们可以定义视频块，使其跨越多个帧中的一个图像块。随后，便可将经过验证的自注意力算法应用于这些词元。最重要的是，可以并行处理多种模态（如文本和图像），从而实现例如根据文本生成图像、根据视频生成文本描述等功能。此外，这些模型可以扩展到规模极大的网络和海量数据集。后续章节将介绍以下多媒体类型。语音识别和文本转语音模型描述了口语与文本之间的相互转换。图像处理的任务是理解图像、用图像描述对其进行描述，并根据文本描述生成新图像。视频理解旨在识别视频中的动作并用文本进行描述。此外，还可以根据文本描述创作新视频。动力系统轨迹刻画了可模拟和控制的序贯决策问题。基础模型可以分析 DNA 和蛋白质序列，以预测相应分子的结构和性质。

关键词： 语音识别、文本转语音、图像描述、文本到图像、视频理解、机器人控制、DNA

基础模型在自然语言任务中取得的惊人成果，促使多媒体处理领域研究其在语音识别和计算机视觉问题中的应用。基础模型最重要的优势之一在于，它们能够对输入序列中元素之间的长距离依赖进行建模，并且与循环网络不同，支持对序列进行并行处理。与卷积网络不同，基础模型在依赖关系建模方面所需的限制极少，并能够定义高维量之间的映射

。此外，基础模型的简单设计允许使用相似的处理模块同时处理多种模态（例如图像、视频、文本和语音）。此外，这些模型能够扩展到规模极大的网络和海量数据集。这些优势推动了使用基础模型对多项多媒体任务取得全面进展。

我们将介绍四个领域中的多媒体应用，并综述当前最佳方法，同时考虑所需资源，例如计算资源和内存开销。

- 语音 识别和文本转语音模型（第 7.1 节）。
- 图像 文本描述和从文本生成图像（第 7.2 节）。
- 视频 理解和视频生成（第 7.3 节）。
- 动态系统轨迹 描述可进行模拟和控制的序贯决策问题（第 7.4 节）。

- *DNA*和蛋白质序列可以使用基础模型进行分析，以预测相应分子的结构和性质。（第7.5节）。

此外，还有许多应用会同时处理多种媒体类型。还有一长列更为专业的媒体类型，已有研究使用多模态预训练语言模型处理这些类型：表格 [25],文本布局 [61], 深度图像[119], 场景图 [60], SQL [18],手语 [199], 点云 [197], 符号知识图谱 [4], 多模态知识图谱 [201],抽象语法树 [202],光流 [50], 等。使用基础模型处理这些媒体类型，与以下各节所述的方法类似。

由于文献中的基础模型数量极其庞大，我们聚焦于撰写时性能较高的代表性模型。我们概述这些方法的内部逻辑和主要特征，同时考虑其所需的资源，例如计算和内存需求。对于标准预训练语言模型，我们提供了指向前面章节中相关描述的链接。Xu等人 [183] 编写了一篇关于使用Transformer进行多模态学习的综述。在“可用实现”标题下，我们列出了该任务可用代码和预训练模型的链接。获取代码的良好来源包括网站 https://paperswithcode.com/、自然语言处理索引https://index.quantumstat.com/以及GitHub https://github.com/github。使用预训练语言模型处理这些媒体类型，与以下各节所述的方法类似。

7.1 语音识别与生成

语音是人类之间最高效、最自然的交流类型。因此，它也是与计算机系统交互的首选类型。在接下来的章节中，我们将介绍用于自动语音识别和文本转语音系统的先进模型。

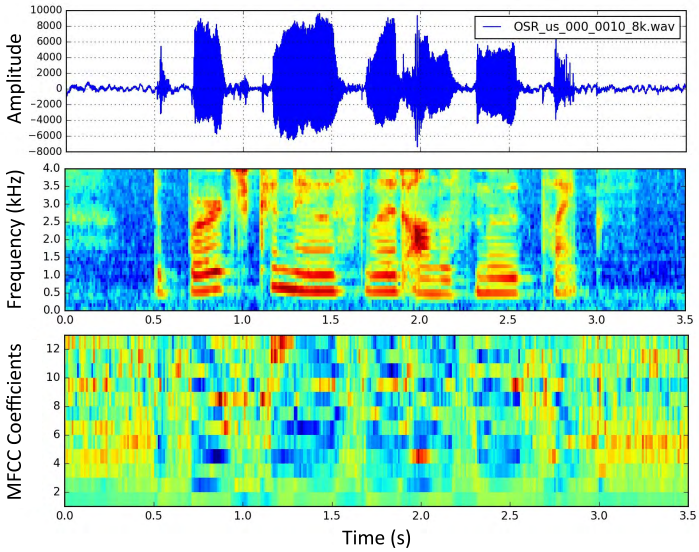


图 7.1 音频信号（上），通过傅里叶变换提取的频率（中）以及相应的梅尔频率倒谱系数（下）。图像来源见表 A.3。

7.1.1 自动语音识别基础

自动语音识别（*ASR*）接收作为音频文件的语音输入，并将其转换为自然语言文本。语音深受性别、社会风格、方言、说话风格和速度的影响。人类语音和口音差异很大，而语音模式中的这些差异是开发自动语音识别系统的主要障碍之一。ASR 开发的另一项阻碍是难以找到足够的训练语料来训练 ASR 模型。目前，在约 6,500 种世界语言中，只有少数语言拥有可用的训练数据。

自 20 世纪 50 年代计算机问世以来，研究人员便开始开发语音识别系统。1984 年，IBM 推出了首个语音识别系统，可识别约 5,000 个不同的英语单词；1993年，消费级自动语音识别产品问世。主要技术包括 n-gram 模型、隐马尔可夫模型和神经网络 [102]。2010年后，基于循环神经网络的语音识别广泛应用于苹果 Siri、亚马逊 Alexa 和 Google 助理等虚拟助手。与此同时，大多数智能手机都内置了自动语音识别功能，即使没有互联网连接，也能通过语音输入文本。

自动语音识别系统最重要的评估指标是词错误率WER = $\frac{S+D+I}{N}$，用于衡量与真实文本的偏差。这里，S 是替换的数量，D 是删除的数量，I 是相较于包含 N 个词语的真实文本，输出中插入的数量。

表格 7.1 主要语音识别技术

模型	机制	性能
ContextNet + NST	目前最佳的基于 LSTM 的自动语音识别系统	LibriSpeech 词错误率 1.7%
Conformer	卷积神经网络 + Transformer 块中的自注意力，作为语言模型的 LSTM	LibriSpeech 词错误率 1.9%
wav2vec 2.0	通过卷积神经网络对语音进行编码，将输入离散化为 Transformer，预测掩码输入。针对语音识别进行微调	LibriSpeech 词错误率 1.5%
组合式自监督学习	Conformer模型 + 无监督 wav2vec 2.0，使用 SpecAugment 生成含噪训练数据	LibriSpeech 词错误率 1.4%
SpeechStew	类似于组合式自监督学习，在 7 个数据集上训练针对语音识别进行微调	LibriSpeech 词错误率 1.7% 不使用语言模型

传统的自动语音识别系统通常由相互独立的部分组成，例如声学模型、发音模型和语言模型。这些部分分别进行训练，然后组合起来进行推理。通常会使用预处理模块来降低音频录音中的信噪比。可以对声音信号应用各种过滤器和方法，以降低相关噪声。此外，还可以使用多个麦克风录制扬声器的声音，从而定位扬声器并大幅降低背景噪声（波束形成） [24]。

随后，特征提取模块的任务是生成与语音识别相关的特征、去除信号中的无关信息并减小输入大小。这通常涉及通过傅里叶变换的变体提取波形的频率。最常用的特征提取方法是梅尔频率倒谱系数 （梅尔频率倒谱系数）、离散小波变换（DWT）和线性预测编码（LPC） [101]。示例见图 7.1。

最后一个模块是分类器，它接收一个固定长度的向量，用于表征给定时间片段内的信号。它会估计下一个时间片段中输出单词或音素的概率。早期的分类器只能处理单个扬声器。后来开发出了能够识别多个扬声器语音话语的新模型。例如，某个自动语音识别系统在Switchboard测试集上实现了 5.1% 的词错误率（WER） [181]。该系统由用于声学建模的卷积神经网络模型（如 ResNet 和 LACE）以及双向长短期记忆网络组成。Malik等人 [101]综述了此前的系统。Papastratis [117], 综述了较新的自动语音识别系统，并讨论了循环神经网络、卷积神经网络和Transformer模型。

7.1.2 基于Transformer的语音识别

基于自注意力的预训练语言模型是序列建模的良好选择，因为它们能够捕捉长距离交互，并且所需的计算开销更小。表 7.1 给出了概览。然而，预训练语言模型不太擅长提取

细粒度的局部特征模式。因此，预训练语言模型与卷积神经网络的组合常用于自动语音识别。目前最佳的基于 LSTM 的自动语音识别系统 **ContextNet +NST** [121] 在 LibriSpeech（clean）上取得了 1.7% 的词错误率。

Conformer [59] 是一种卷积增强Transformer。Conformer将卷积层模块（第1.7节）和自注意力模块（第2.3节）作为层集成到编码器块中。卷积模块包含一个门控线性单元 （GLU）激活层，通过扩展因子为2的 1×1 逐点卷积对通道数量进行投影，从而选择对预测重要的特征。随后是一维深度卷积，它为每个输入通道应用一个单独的卷积滤波器。之后进行批归一化，然后是一个*Swish* [131] 激活层。

最终得到的模型包含17个Conformer块，参数量最多达1.18亿，并在包含不同扬声器朗读的有声读物的*LibriSpeech* [116] 数据集上进行训练。该模型针对每个10毫秒的时间片获取一个包含80个滤波器组特征的向量（图7.1）。作者使用 SpecAugment [120] 掩码输入信号的不同部分，以对模型进行正则化。此外，他们还在LibriSpeech语料库上训练了一个3层LSTM语言模型，用于预测下一个词语。语言模型的输出与Transformer输出相结合，以突出显示句法和语义上正确的词语。结合LM后，Conformer在LibriSpeech（clean）上的WER达到1.9%。不使用LM时，WER为2.1%。

该**S4** [58] 模型能够处理最多包含 16k 个元素的长输入序列（第 3.2.2 节）。该模型被应用于语音分类，在处理原始语音信号时，将最先进水平提升至 98.3%。与此前 95.3% 的最先进准确率相比，错误率大幅降低。预计该模型也能显著减少其他语音识别任务中的错误。

7.1.3 语音识别自监督学习

语音自监督学习有望借助更多未标注数据提升语音识别结果。研究表明，在大量未标注数据上进行自训练，能够显著提升模型性能，使模型仅需相对少量的微调数据便可取得优异表现 [184]。

wav2vec2.0 [10]在语音数据上进行无监督学习，无需文字转录。与用于文本的 BERT模型类似，它通过预测被掩码的声音“词元”进行学习。wav2vec 使用多层卷积神经网络对原始语音音频进行编码，为每个时间点生成语音的潜在表示。随后，量化模块将连续的潜在表示离散化为词元 $\boldsymbol{q}_t$。这种离散化操作不连续，会阻碍梯度反向传播。

一种解决方案是在采样的离散结果与概率分布之间进行插值。这可以通过 *Gumbel-Softmax* 分布 [75]实现。要以概率 $p_1, \ldots, p_k$对离散分布进行采样，我们可以抽取一个随机均匀变量 $U \sim \text{uniform}(0,1)$，并计算 $Z = \text{onehot}(\max_i p_1 + \cdots p_{i-1} \leq U)$，其中 $i = 1, \ldots, k$ 是离散索引，而

onehot(j)会生成一个全为零、在位置 j处为一的向量。由于使用了 max 函数，这种采样不可微。另一种公式是

$$Z = \text{onehot}(\text{argmax}_i(G_i + log(p_i))), \tag{7.1}$$

其中， $G_i \sim \text{Gumbel}(0, 1)$ 是从标准 Gumbel 分布中抽取的独立同分布样本。这将 Z 的采样重构为参数与某种具有固定分布的独立噪声所确定的函数。现在可以使用 softmax 函数作为 argmax 的可微近似：

$$y_i = \frac{\exp(G_i + \log p_i)/\tau)}{\sum_j \exp(G_j + \log p_j)/\tau)}. \tag{7.2}$$

τ 是温度参数，用于控制新样本对离散向量的近似程度。这种近似在训练期间使用，而离散化的独热编码向量则在评估期间计算。wav2vec 通过这种方法计算离散向量$\boldsymbol{q}_t$ 。

随机采样的连续 10 个时间步的 $\boldsymbol{q}_t$ 表示会被掩码，并必须通过类似于 BERT 的 Transformer 进行重建。自注意力捕获整个潜在表示序列中的依赖关系。该模型在超过 1000 小时的标注和未标注语音数据上进行了预训练。通过在上下文网络顶部添加一个随机初始化的线性投影，将预训练模型微调用于语音识别，该投影映射到 C 个类别，这些类别包括字符以及词边界标记。为了处理跨越多个时间槽的字符，采用了 连接时序分类 （CTC）损失 [57] 。微调使用了 5 小时带有音素标注的音频数据。在 LibriSpeech 上，作者取得了 2.1% 的 WER。另一个使用 3 亿个参数的类似模型，在 wav2vec 训练中使用 5.3 万小时的未标注数据，并在微调中使用 1000 万条标注数据，在 LibriSpeech 上取得了 3.0% 的 WER [184]。使用全部数据进行训练可将 WER 降低至 1.5%。

组合式自监督学习 [196] 将 wave2vec 无监督预训练与 Conformer 相结合。自动语音识别网络是一个序列“翻译器”，由参数量高达 10 亿的 Conformer 编码器和多层 LSTM 解码器组成。此外，作者采用噪声学生训练（NST），通过教师模型对音频进行推理，为未标注数据生成转录文本。经过筛选和平衡后，这些教师标注数据会用于训练下一代自动语音识别模型。在 LibriSpeech 上，该模型以 1.4% 的词错误率达到最先进水平。

w2v-BERT [31] 一方面通过对比学习，将连续的语音信号离散化为有限个具有判别性的语音词元；另一方面，模型通过执行以离散化词元为输入的掩码预测任务，学习上下文化语音表示。预训练期间，这两项任务以端到端方式同时优化。微调期间，预训练的 w2v-BERT 模型（拥有 1B 个参数）的输出由长短期记忆网络解码器汇总。在 LibriSpeech 基准（test-clean）上，其词错误率为 1.4%，与领先系统相近；在 LibriSpeech 基准（test-other）上，该模型的词错误率达到最先进水平，为 2.5%。此外，该模型的 600M 参数版本经过微调后，应用于

一项语音搜索任务，用户可以通过在手机或计算机上说话来使用谷歌搜索。该任务由语音摘要片段组成，平均时长为 5.5 秒。模型将错误率降低了约 30%，达到 6.2。**SpeechStew** [21] 采用结合 wav2vec 预训练的 Conformer1B。它使用 7 个可用的语音识别数据集进行预训练，没有进行任何依赖领域的重新平衡或重新加权。不使用语言模型时，它在 LibriSpeech 上的词错误率为 1.7%。

TERA [98]是一种自监督语音模型，采用多目标辅助任务，在大规模未标注语音训练集上预训练 Transformer 编码器。输入可以是任意声学特征，例如梅尔频率倒谱系数。模型通过从经过修改的样本中重建声学帧来学习，这些样本在三个性质上经过随机改变：时间扰动要求从受损的时间步区块中进行重建。通道扰动要求从缺失的频率通道区块中恢复信号。幅度扰动则涉及重新生成经过改变的特征幅度。通过重建这些数据变化，模型能够学习到更好的上下文化表示。时间扰动的宽度设为 85 毫秒语音，约为平均音素时长。与 BERT 类似的最大模型有 1.7 亿个参数。该模型在音素分类、说话人识别和语音识别方面表现出色，例如在 TIMIT 基准测试中，音素错误率（PER）为 14.5%。

在一项全面的分析中，Zhang 等人 [195] 评估了自监督预训练对 ASR 的益处。他们采用参数量为 6 亿至 80 亿的 Conformer模型，这些模型在包含数千至百万小时音频的超大规模、多样化无标签数据集（*BigSSL*）上进行预训练和自训练。仅使用 3% 的标注数据，他们便取得了与语音搜索基准测试的最先进水平相当的结果。在八项 ASR 基准测试中，经过预训练后，他们能够达到或超越最先进水平。在语言识别和情绪检测等五项非 ASR 任务上，他们也能提升最先进水平。对于大型数据集，预训练带来的增益较小，但仍然显著。

许多应用不仅受益于理解词语，还受益于理解其他信息，例如一个人在一段话语中的情绪、扬声器是否佩戴了掩码，或语音是否为合成语音。Shor [156] 提出了一种基于 Conformer 的大规模架构，参数量超过 6 亿，可通过微调来检测这些额外特征，并取得最先进水平的性能。

可用实现

- Conformer：https://github.com/PaddlePaddle/PaddleSpeech
- wav2vec：https://github.com/facebookresearch/fairseq 用于翻译、摘要生成、语言建模和其他文本生成任务的序列建模工具包。
- <风格 id="2">TERA：</风格>https://github.com/s3prl/s3prl
- Hugging Face 语音识别：https://huggingface.co/models?pipeline_tag=automatic-speech-recognition
- TensorFlow 语音到文本：https://tfhub.dev/s?module-type=audio-stt

7.1.4 文本转语音

语音合成是指从文本、唇部动作等其他模态生成语音。文本转语音（文本转语音）系统旨在将自然语言文本转换为语音。平均意见分（平均意见分）是评估生成语音质量最常用的方法。平均意见分定义为人工评估者在主观质量评测中对给定刺激进行单项评分的算术均值。平均意见分的范围为 0 到 5，真实人类语音的得分在 4.5 到 4.8 之间。Tan 等人提供了一份全面且最新的文本转语音系统综述。 [163]

早期的文本转语音系统只是简单地拼接预录语音片段，而现代系统则会对语音进行完整合成。**WaveNet** [114] 是首个成功对音频信号的原始波形，而非声学特征进行建模的模型。它能够以每秒 16,000 个样本的速率生成新的类语音波形。WaveNet 的核心是一个自回归模型，由扩张卷积组成，其中每个样本都依赖于前面的样本。在每一层中，所包含的时间步数量都会翻倍。WaveNet 将平均意见分从 3.86 提高到了 4.21。*Fast WaveNet* 通过缓存先前的计算结果，将二次时间复杂度降低为线性复杂度。

Tacotron 2 是一种直接根据文本进行语音合成的神经网络架构。它由一个带有注意力机制的循环长短期记忆网络序列到序列特征预测网络组成，该网络根据输入字符序列预测一系列梅尔频谱帧；同时还包含一个经过修改的 WaveNet，根据预测的梅尔频谱帧生成以其为条件的时域波形采样。Tacotron 2 获得了令人印象深刻的 4.53 平均意见分。

由于TTS执行类似于NLP的序列处理，因此在这一领域使用PLM也是很自然的。基于Transformer的模型旨在缓解以Tacotron 2为代表的早期TTS方法的两个问题：训练和推理的计算成本高，以及使用LSTM对长距离依赖进行建模的困难。

Transformer TTS [94] 将原始Transformer编码器-解码器 [168]应用于语音合成。编码器接收音素作为输入，这些音素经过由CNN和全连接层组成的编码器预网络进行处理。标准Transformer编码器输出上下文音素嵌入（图 7.2）。解码器接收梅尔帧作为输入，这些输入经过包含两个全连接层的解码器预网络转换，以生成适当的嵌入。标准解码器生成梅尔帧输出嵌入。随后，这些嵌入经过两个不同的线性投影，分别用于预测梅尔频谱图和停止标记。一个5层CNN生成残差，以优化梅尔频谱图的重建。WaveNet声码器生成最终的音频输出。Transformer的编码器和解码器均由6层和8个注意力头组成。该模型的速度约为Tacotron 2的4.25倍，其平均意见分为4.39，接近人工质量。

FastSpeech 2 [138] 解决了这样一个问题：由于语音中的音高、时长、音量和韵律等变化，同一段输入文本可能对应多种不同的语音序列。它通过 Transformer 编码器对输入音素进行编码，生成嵌入。随后，方差适配器将时长、音高等不同的方差信息加入隐藏序列中，

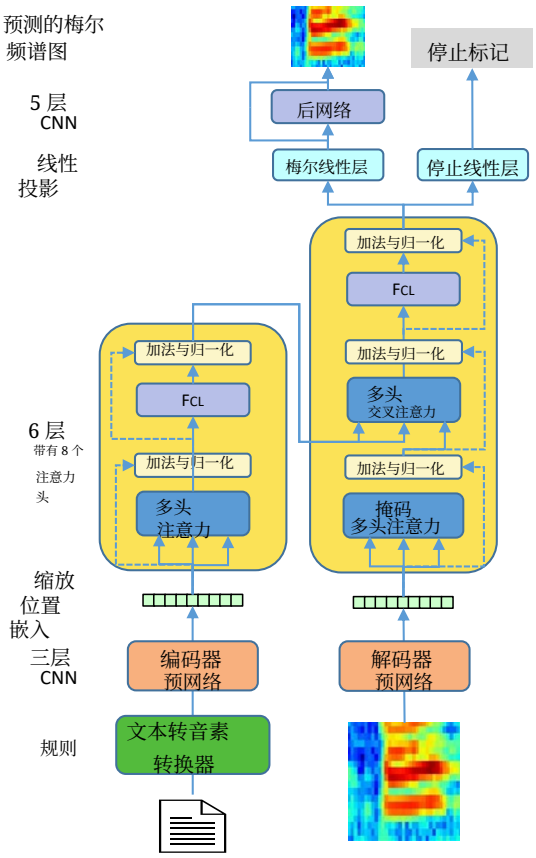


图 7.2 使用 Transformer TTS 进行语音合成。编码器和解码器均包含 6 层、8 个注意力头以及残差连接。由此得到的梅尔频谱图通过 WaveNet 声码器转换为最终的音频输出 [94]。图像来源见表 A.3。

例如时长、音高和能量。最后，梅尔频谱图解码器将调整后的隐藏序列并行转换为梅尔频谱图序列。编码器和梅尔频谱图解码器都包含由 Transformer 块和一维卷积构成的层。方差适配器不仅预测时长，也预测音高和能量；它使用包含一维卷积、前馈层和层归一化的层，并通过 Dropout 进行正则化。

变体 *Fastspeech 2s* 可直接从文本生成波形，无需依次生成梅尔频谱（声学模型）和波形（例如使用 wav2vec 这类声码器）。最终的波形解码器由门控激活以及不同类型的一维卷积和膨胀一维卷积组成，以覆盖更广的时间范围。作者在波形解码器中采用对抗训练，迫使模型自行隐式恢复相位信息。

在实验中，作者得出了以下平均意见分（MOS）值：Tacotron 2：3.70，Transformer TTS：3.72，FastSpeech 2：3.83，FastSpeech 2s：3.71，人类语音：4.30。请注意，与人类语音之间的差异主要是由

声码器造成的。此外，在推理时，FastSpeech 2 和 FastSpeech 2s 的速度约为 Transformer TTS 的 50 倍。

AdaSpeech 2 [186] AdaSpeech 将文本转语音系统适配到目标说话人。只需目标说话人的音频录音，无需文本转录。作者将梅尔频谱编码器应用于经过充分训练的文本转语音模型，以进行语音重建，同时约束梅尔频谱编码器的输出序列，使其接近原始音素编码器的输出序列。梅尔编码器也由 4 个前馈 Transformer 模块组成。需要注意的是，无需重新训练原始系统，只需训练梅尔编码器。在针对目标说话人进行微调时，会调整梅尔解码器的参数。该模型的语音质量平均意见分与使用转录文本进行适配的文本转语音系统不相上下。

最近，亚马逊宣布 Alexa 将能够模仿其他人的声音 [17]。为了“让回忆永存”，Alexa 可以用已故祖母的声音讲故事和播放音乐。亚马逊指出，只需约一分钟的音频录音就能模仿一种声音。

可用实现

- Tacotron 2： https://github.com/NVIDIA/tacotron2
- TransformerTTS： https://github.com/as-ideas/TransformerTTS
- FastSpeech2: https://github.com/ming024/FastSpeech2
- AdaSpeech2: https://github.com/rishikksh20/AdaSpeech2
- Hugging Face TTS： https://huggingface.co/models?pipeline_tag=text-to-speech
- Mozilla TTS 面向所有人的文本转语音： https://github.com/mozilla/TTS
- TensorFlow TTS： https://tfhub.dev/s?module-type=音频语音合成

7.1.5 语音到语音语言模型

GSLM [89] 是一种语言模型，它接收原始语音音频作为输入并直接生成输出。例如，它可以用于创建一个无需中间文本表示的对话系统。在内部，该模型会将输入的原始语音转换为离散伪文本单元。作为离散化器，使用了 CPC [113], wave2vec 2.0[10], 和 HuBERT [68] 来创建不同长度（50、100、200）的嵌入。由于没有声音单元词汇表，而且声音单元长度不一、没有明显的分段方式，因此很难选择单元。与 BERT 类似，HuBERT 使用掩码预测任务进行训练，输入为经过掩码的连续音频信号。在实验中，HuBERT 在大多数情况下表现最佳，其次是 CPC。

自回归“基于单元”的语言模型包含 12 层，使用由 6,000 小时的*LibriLight* 语音数据 [139]生成的样本进行训练，每个样本最多包含 3,000 个单元。为了将单元转换为语音，研究人员采用了经过修改的 Tacotron-2 模型[154]，该模型以伪文本单元作为输入，并输出对数梅尔

频谱图。为了生成波形，研究人员使用了预训练的声码器*WaveGlow* [125]，它可将对数梅尔频谱图转换为语音。

在首次测试中，语音输入被编码为若干单元，再转换为语音。研究人员通过人工平均意见分（MOS）评估生成语音的可懂度。在使用 *LJ Speech*数据 [74] 进行训练时，无监督模型的 MOS（平均意见分）得分为 4.00，而自动语音识别（ASR）与文本转语音（TTS）系统的组合得分略高，为 4.04 [89]。在测试完整的语言模型生成时，该模型的 MOS 得分为 4.01，而自动语音识别与语言模型的组合得分为 3.91。据作者称，生成的语音听起来像英语，音素和词语也可以辨认。示例表明，在语言和句法层面仍需改进。用于声音转录时，200 个单元效果良好；而用于语言建模时，较少的单元似乎更合适。预计增加训练数据可以提升质量。

7.1.6 音乐生成

预训练语言模型也可应用于其他序列数据，例如音乐。一方面，可以训练音乐语言模型，使其能够生成与训练数据相对应的新音乐。另一方面，模型也可以根据外部信息（例如歌词或视频）生成音乐。Bilici [14] 对近期的音乐生成模型进行了综述。

一种突出的音乐生成方法是 **MuseNet** [123]，它采用了 GPT-2 的变体 Sparse Transformer。它会在长度为 4,096 个 MIDI 字符的上下文中计算注意力模式。要生成新的乐曲，可以选择一位作曲家，并使用某首已知作品的开头音符。接着最多可以选择十种不同的乐器，系统便会生成一首符合所需特征的乐曲。专家给出的评价相当正面。同样，**Music Transformer** [71] 能够生成钢琴曲。**主题 Transformer** [155]以一个主题作为输入，并经过训练，使其在生成结果中多次包含该主题。

Jukebox [36] 采用多尺度向量量化器变分自编码器模型（VQ-VAE） [113]，将原始音频压缩为离散代码。该模型基于自回归 Transformer，也适用于人声。模型采用了三个具有不同时间分辨率的独立 VQ-VAE 模型。训练好的模型可以根据艺术家和音乐风格进行条件控制，以引导音乐和人声风格；也可以根据未对齐的歌词进行条件控制，使歌唱更易于控制。该模型能够生成长达数分钟的片段，并以听感自然的声音生成具有辨识度的歌唱。 [35]现已提供多个样本。

CMT [38] 为特定视频生成背景音乐。其目标是匹配视频的节奏、时间安排和运动速度。CMT 从视频中提取这些特征，并支持对音乐风格和乐器进行全局控制。该模型不需要成对的视频和音乐训练数据。实验表明

生成的背景音乐与输入视频达到了令人满意的兼容性，同时还具有出色的音乐质量。

可用实现

- CMT 可控音乐 Transformer https://github.com/wzk1015/video-bgm-generation
- Jukebox：一种音乐生成模型 https://github.com/openai/jukebox

7.1.7 总结

近年来，语音识别取得了巨大进展，基础模型如今已成为这一任务中成熟的方法。它们与卷积神经网络模块相结合，能够捕捉长距离交互并缩短处理时间。与自然语言处理类似，自监督学习带来了显著的性能提升。与自然语言处理中使用词元不同，这些模型会生成离散声音表示。许多不同的模型都遵循这一方案，并能够提升在不同基准测试上的最先进水平。

近年来，从文本生成语音的技术取得了显著进步。WaveNet 是首个能够以每秒 16,000 个样本生成类语音波形的模型。Transformer 可用于将输入音素转换为梅尔频谱图，声码器再根据这些频谱图生成语音音频。此外，还有 FastSpeech 2s 等变体，可以直接将文本转换为音频信号。这些模型的输出质量已接近人类语音。一些模型能够使其输出适应不同扬声器的声音。这令人印象深刻，但如果利用这种方式以某人的声音生成虚假话语，也会带来严重的安全问题。近期用于长输入序列的 S4 状态空间模型在语音信号分类方面将错误减少了 60%。可以预期，该模型也将大幅减少其他语音识别任务中的错误。

语音识别和文本转语音可以与其他应用相集成。SpeechBert [30] 是一个端到端的语音问答（SQA）模型，它使用单个 Transformer 编码器对音频和文本进行编码，并在语音和文本语料库上使用 MLM 进行预训练，然后针对问答进行微调。实时语音翻译会在智能手机上即时生成，从而支持使用外语进行无缝交流 [78, 81]。而 GSLM 是一种生成式语言模型，可以直接处理离散化声音词元。

音乐生成是一个相关主题。自回归预训练语言模型（例如 MuseNet 或 Music Transformer）可以基于大规模语料库的预训练来生成音乐。此处可以选择作曲家风格和乐器。此外，还可以根据某些输入来生成音乐，例如 Jukebox 模型使用的歌词文本，或用于创作背景音乐的视频。

7.2 图像处理与生成

预训练语言模型在自然语言处理领域取得的突破，引发了计算机视觉社区的极大兴趣，促使研究者将这些模型应用于视觉和多模态学习任务。Transformer 成功的两个重要因素是自注意力和自监督。自注意力层会生成表示，并将词元（文本词元和/或视觉词元）之间的关系纳入考量。自监督则在大规模数据集的训练过程中预测数据元素中被掩码或修改的部分。它无需人工标注即可从数据中获取大量知识，而且与卷积神经网络和循环神经网络等其他模型相比，所需的归纳偏置较少。Khan 等人 [84]和 Du 等人 [43]对用于视觉与语言应用的预训练语言模型进行了全面综述。Hafiz 等人 [62] 概述了注意力机制和机器视觉深度学习。近期还有一篇关于视觉与语言研究的教程 [6]。本节讨论的模型的主要特征汇总于表 7.2。

7.2.1 图像处理基础

如图 7.3 所示，图像处理可以解决各种任务。通过对图像中最重要的物体进行分类，可以描述图像的主要内容。更具挑战性的是识别并分类图像中的相关物体，这还需要用边界框描述物体的位置。生成图像描述时，需要识别图像中最重要的物体及其关系，并用自然语言句子加以描述。为图像创建描述，需要识别图像中最重要的物体及其相互关系，并用自然语言句子加以描述。与此相关的任务是检索与某个图像描述相对应的图像。视觉问答需要理解问题并分析图像，以自然语言生成回答。多模态验证是其中的一种变体，需要判断有关图像的陈述是否属实。

许多任务都涉及创作一幅新图像。一个典型示例是根据图像描述生成一幅全新的图像。另一种做法是填补图像中的缺失区域。还有一种变体是根据图像描述改变图像的风格，例如将照片转换成梵高风格的绘画。这也可以针对特定的图像区域进行。

图像如何表示以供 Transformer 使用，是一个重要方面。语言模型会将文本划分为词元序列，作为 Transformer 的输入。图像也采用相同的方法进行处理，即划分为较小的图像块。每个图像块的内容都可以用一个向量表示，该向量构成 Transformer 的输入。图像块的位置则通过位置嵌入进行编码，并将其加到输入嵌入上。

图像块的嵌入可以简单地通过对其像素值进行可学习的线性变换来实现。也可以使用其他变换，例如小型卷积神经网络模型或



图 7.3 图像分析可用于解决多种不同的任务。根据任务的不同，系统会接收文本（绿色）和图像作为输入，并生成文本（蓝色）和图像作为输出。图像来源见表 A.3。

变分自编码器（第 1.7 节）。为了获得更稳健的表示，通常会对生成向量进行离散化，以去除局部噪声。此外，也可以将图像描述或区域标注中的文本作为输入。与通常情况一样，这些文本会被转换为词汇表中的词元。

为对图像元素与文本之间的交互进行建模，可以采用不同的 Transformer 架构（表 7.2）。单流架构 会将所有输入拼接起来，并由单个 Transformer 进行处理。这样可以确定不同输入元素之间的交互，但需要处理长序列。双流或多流架构 使用独立的预训练语言模型处理不同模态或图像分辨率。在这种情况下，输入序列较短。人们提出了流之间的多种交互形式（例如交叉注意力）。之后，可以通过相似度度量比较输出，或由其他预训练语言模型将其组合起来。

视觉预训练任务遵循文本 Transformer 的模式。掩码语言建模 (MLM)会对部分输入词元进行掩码，并要求模型根据上下文预测这些词元。如果同时存在文本和图像词元，则可以利用这两种模态中的信息来执行此任务，模型也会学习文本与图像元素之间的关联。类似地，也可以对图像区域进行掩码，并根据文本和图像上下文重建这些区域。在分类任务中，模型可以判断图像描述是否准确描述了某张图像，或只是随机文本。通过这种方式，可以训练文本与图像之间的相关性。另一个目标是在同一语义空间中学习图像与单词的联合表示：将匹配的图文对的嵌入拉近，同时推开

表 7.2 文本与图像结合的主要技术。**基准测试：** 视觉问答：COCO视觉问答数据集（第 7.2.5 节） [56]；图像生成：经微调的 MS-COCO图像生成基准测试；img-gen-0：MS-COCO图像生成基准测试零样本；ImageNet：ImageNet 分类 top1准确率；字幕：MS-COCO图像描述基准测试；FID：Fréchet Inception距离应较小（第 7.2.6 节） [64]。括号中的数字为参数数量。

模型	方法	基准测试
视觉Transformer（ViT）第 7.2.2 节	拼接文本词元与由以下内容生成的图像词元图像块。使用 BERT自编码器进行处理，并执行分类（632M）。	ImageNet最先进准确率 90.5%
CLIP 第 7.2.4 节	使用视觉Transformer编码图像，并使用 GPT 自编码器编码文本。最大化图像与嵌入的相似度，预测它们是否属于同一组。	
VilBERT 第7.2.5节	使用 Faster R-CNN 提取边界框。图像区域和文本由两个 BERT 自编码器进行编码并执行交叉注意力。针对视觉问答进行微调	视觉问答最先进水平 70.9%
OSCAR 第 7.2.5 节	使用 Faster R-CNN 提取边界框。BERT 自编码器将区域描述与文本关联起来。针对 7 个任务进行微调，例如图像描述	字幕 最先进水平 41.7 BLEU-4
VinVL 第 7.2.5 节	使用 ResNeXT 模型作为区域提取器，并使用 OSCAR。针对图像描述进行了微调	字幕 40.4 BLEU-4
DALL-E 第 7.2.6 节	文本被编码为词元，图像被转换为图像词元（通过变分自编码器（VAE））。使用 GPT-3（120亿参数）生成新的图像词元。	img-gen-0 17.9 FID
GLIDE 第7.2.7节	逆转会破坏图像的扩散过程。生成通过 U-Net模型（3.8B）对图像进行微小改动。	img-gen-0 最先进水平 12.2 FID
XMC-GAN 第 7.2.7 节	基于 GAN 的图像生成器，生成器创建图像，判别器区分真假图像。	图像生成领域最先进水平 9.3 FID
CogView 第 7.2.7 节	向量量化变分自编码器。GPT 模型（4B）使用以下内容进行训练：文本词元和量化图像词元	模糊图像上的图像生成领域最先进水平
LAFITE 第 7.2.7 节	使用 CLIP 将文本转换为图像嵌入。训练调节 StyleGAN2 的层以生成图像	图像生成领域最先进水平 8.1 FID img-gen-0 16.9 FID
OFA 第 7.2.8 节	使用文本、图像词元以及带有边界框的物体。序列到序列模型（472M），经过预训练以关联词元和物体。文本指令控制 9 种不同的任务	图像生成领域最先进水平：FID 10.5 字幕领域最先进水平：BLEU-4 43.5
DALL-E 2 第 7.2.7 节	通过 CLIP 从文本生成图像嵌入，通过扩散解码器转换为 1024×1024 图像	img-gen-0 最先进水平 10.4 FID
Imagen 第 7.2.7 节	通过 T5-XXL 生成文本嵌入，生成图像块通过扩散模型生成，上采样至 $1,024 \times 1,024$ 通过两个超分辨率扩散模型	img-gen-0 最先进水平 7.3 FID
Stable Diffusion 第 7.2.7 节	使用 U-Net 和扩散生成图像	ImageNet 条件下的 3.6 FID

不匹配的图文对。对于这种图像到文本的对比损失，嵌入之间的接近程度通过嵌入之间的标量积来衡量。

7.2.2 视觉Transformer

该**ViT** （视觉Transformer） [42] 将纯Transformer编码器（第2.3.1节）应用于图像块。输入图像 $\boldsymbol{x} \in \mathbb{R}^{H \times W \times c}$ 包含 $H \times W$ 个像素和 c 个颜色通道。它被划分为像素大小为 $s \times s$ 的图像块，例如 $s = 16$。每个$N = HW/s^2$ 图像块由 $s^2 * c$ 个数字组成，这些数字经过线性映射后变为长度为 d 的向量，作为Transformer的输入。通常会添加一维

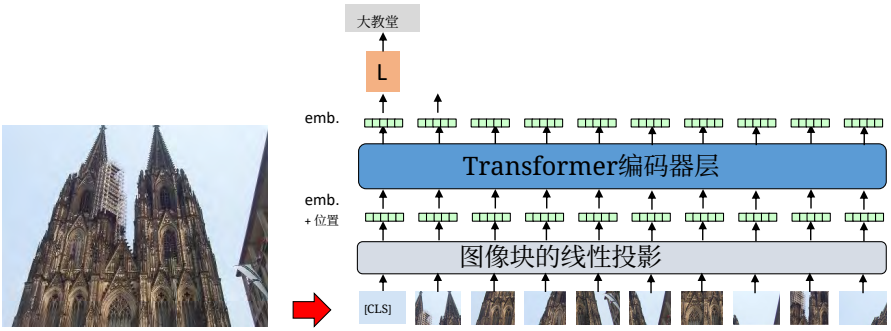


图 7.4 视觉Transformer ViT 将图像划分为固定大小的方形图像块。通过线性投影为每个图像块计算嵌入。标准编码器计算上下文嵌入。使用 [CLS] 词元的嵌入，通过逻辑分类器[42]计算一个类别。经作者许可改编图像，鸣谢信息见表 A.3。

位置嵌入，因为二维位置并未显著提升性能。本文采用了不同的模型：ViT_{Base}、$\text{ViT}^{\text{Large}}$和$\text{ViT}^{\text{Huge}}$，它们分别具有12、24和32层，以及8600万、3.07亿和6.32亿个参数。

Transformer 编码器的输入序列长度为 N，由大小为 d的向量组成。每一层生成长度为 d的 N 个嵌入。最后一个编码器块中 [CLS] 词元的输出嵌入作为逻辑分类器的输入，用于计算图像类别的概率。该架构如图 7.4 所示。

值得注意的是，图像可以使用不同的输入图像分辨率进行训练。但图像块大小始终相同，因此输入大小会有所不同。为适应新的分辨率，需要对位置嵌入执行二维插值。模型通常在大型数据集 JFT-300M [161] 上进行预训练，然后使用不同的分类器层在较小的任务数据集上进行微调 [189]。在比分辨率预训练更高的分辨率下进行微调通常有益。模型在最多包含 3 亿张图像的数据集上进行了预训练。

最大的模型 ViT^{Huge} 的输入图像块大小为 14×14。它在 ImageNet 上击败了改进版的预训练 ResNet152 [63] （具有 152 个 CNN层）和 EfficientNet [92]，并在将图像分类为 1,000 个对象类别 [118]的任务中取得了 90.5% 的最先进水平 Top-1 准确率。在 ImageNet 测试集上，预训练使绝对准确率提高了 13%。借助 2.5k 个 TPUv3 日，其所需的计算开销（包括预训练）仅为 ResNet 所需计算开销的 25%。它还在另外 5 个热门图像分类基准上刷新了最先进水平。较小的 $\text{ViT}^{\text{Large}}$ 输入图像块大小为 16×16，同样击败了 ResNet152，而所需的计算开销仅为 ResNet152 的 6.8%。

当 ViT 在 ImageNet 这类中等规模数据集上训练时，在参数数量相近的情况下，其性能低于 ResNet（第 1.7 节）。这似乎是因为卷积神经网络具有更合适的归纳偏置，例如平移等变性和局部性，而 Transformer 必须通过预训练来学习这些特性。



图 7.5 输入图像显示在上排。下排描绘了视觉Transformer模型为进行分类而针对输入空间计算出的主要注意力区域。经作者[42,p. 8]友好许可转载。

因此，只有预训练的 Transformer 才能胜过卷积神经网络，但这需要较低的计算开销。Cao 等人 [20] 介绍了一种方法，说明如何利用有限数据训练视觉 Transformer 并取得良好结果。Chefer 等人 [22]介绍了一种基于泰勒分解方法的新方法，用于可视化促成某一图像分类结果的图像区域。

分析训练后模型的内部结构很有助益。事实证明，训练得到的位置嵌入反映了输入图像的行列结构，位于同一行或同一列的图像块具有相似的嵌入。根据注意力权重，可以确定特定注意力头所关注的图像部分。一些注意力头会考虑整幅图像，而另一些注意力头在较低层中始终具有较小的注意力距离。这可能与卷积神经网络中的早期卷积层具有相似的功能 [130]。一项实验研究表明，Transformer对严重遮挡具有很强的鲁棒性 [108]。与卷积神经网络不同，卷积神经网络通常依据纹理而较少依据形状来检测宾语；在形状识别方面，ViT与人类的表现相当。图 7.5 展示了整个ViT模型对应于语义相关区域的注意力区域。

许多研究人员对 ViT 的稳健性进行了研究。在一系列实验中，Mao 等人 [103]发现，ViT 倾向于使用包含纹理和噪声的局部特征，并在一定程度上忽略形状和结构等全局上下文。对此，他们提出使用基于变分自编码器（*VQ-VAE*）的向量量化器，将连续的输入特征离散化为图像词元 [113]。他们报告称，在多个 ImageNet 分类基准上，准确率最高提升了 12%。 [146]提出了一种类似的、用于 ViT 的自适应词元生成方法。**BEiT** [11]采用受 BERT（掩码图像建模）启发、基于 VQ-VAE 的自监督方法，其性能优于监督式预训练 ViT。



图 7.6 ColTRan [88]对灰度图像（左）进行的不同着色效果。请注意，语义约束通常都能得到遵循，例如皮肤和树叶的颜色。经作者 [88, p. 1]慷慨许可，转载此图。

7.2.3 图像生成

此外，还有许多用于各种图像增强任务的预训练语言模型。图像超分辨率 将低分辨率图像转换为更高分辨率的图像。**SwinIR** [96]基于层次化表示构建：从小尺寸图像块开始，并在更深的层中逐步合并相邻的图像块。在训练过程中，预训练语言模型将经过 CNN 层预处理的小尺寸图像作为输入。Transformer 块包含 Transformer 层和 CNN 层，并经过训练以重建高分辨率图像。SwinIR 在超分辨率、图像去噪和 JPEG 压缩伪影消除基准测试上达到了最先进水平，同时仅具有 1200 万个参数。

ColTran [88] 使用带有列注意力和行注意力的 Transformer，将灰度图像转换为全彩色图像。它首先通过条件Transformer，为经过空间降采样且仅包含 512 种粗略颜色的图像预测颜色。随后，两个全并行 Transformer 将粗略着色的低分辨率图像上采样为全彩色高分辨率图像。与其他方案相比，该模型在 ImageNet 数据上取得了最佳 FID 分数 19.7。图 7.6 展示了着色示例。

SwinTransformer [99] 从较小的图像块开始，并在更深的 Transformer 层中逐步合并相邻图像块，从而构建图像的层次化表示。通过在划分图像的、大小为 7 的不重叠窗口内局部计算自注意力，实现了线性计算复杂度。在相邻层之间，自注意力窗口会发生移位，使其与前一自注意力层中的相邻窗口重叠。最大版本的模型包含 197M 个参数，可处理分辨率为 384×384 的图像。在 ImageNet 分类任务中，该模型的 Top-1 准确率达到 87.3%。在图像目标检测任务中，Swin Transformer 也能够提升此前的最佳结果。

VQ-GAN [45] 使用卷积神经网络高效地学习由丰富上下文的视觉块组成的码本，随后学习这些视觉块的全局结构模型。这些图像块内部的长程交互需要一个具有强大表达能力的 GPT-2 来对视觉块的分布进行建模。图像块字典根据感知损失 [41, 194]捕捉感知上重要的局部结构。该损失通过



图 7.7 VQ-GAN [45] 使 Transformer 能够合成像素为 1280x460 的高分辨率图像。经作者 [45, 允许转载。第 12873]页。

一种对抗训练过程进行优化，该过程使用基于图像块的判别器来区分真实图像和重建图像。

一个含有 307M 个参数的 GPT-2 模型经过预训练，用于生成图像语料库中编码图像的代码序列。每幅图像都被划分为 16×16个图像块，序列长度为 1024。图 7.7 展示了一个示例图像。如果训练语料库包含类别信息 c，则可以生成特定类别的图像。类别信息也可以限定到特定图像区域。VQ-VAE 在重建 ImageNet 照片时的 FID 约为 10，而 VQ-GAN 的表现好得多，得分为 1.7。

StyleSwin [191] 是 VQ-GAN 的进一步发展。它采用了上文讨论的*Swin Transformer* [99]。StyleSwin 在频域中使用小波判别器来抑制块状伪影。这个含有 41M 个参数的模型在多个成熟的基准测试中达到了最先进的质量水平。图 7.8 展示了示例图像，其整体几何结构连贯，且具有高保真细节。在 CelebA-HQ 1024 基准测试中，StyleSwin 的 FID 为 4.4，优于此前所有模型，包括 StyleGAN2（5.1）。对于基于 LSUN 数据集生成教堂图像这一任务，StyleSwin 的 FID 分数为 3.1，几乎与得分最高的对抗式 CIPS 模型 [7]（FID 分数为 2.9）不相上下。

Data2vec [9] 提出了一种新的自监督学习训练准则，可应用于图像、文本和语音数据。它包含两种模型：教师模型，用于处理完整输入；学生模型则在掩码部分数据的同时处理输入。

该模型采用标准Transformer架构，并针对不同媒体使用特定的输入编码。图像通过线性变换后的图像块进行编码，方式与ViT类似。语音数据通过多层一维卷积进行编码。文本数据则编码为子词标记。学生模型的训练目标由教师网络顶部 K 个编码器块的平均表示构建；教师网络会处理完整输入。学生模型只能接收掩码后的输入，因此必须预测这一目标。Data2vec 的表示是连续的，并通过上下文信息进行建模



图 7.8 由 StyleSwin 基于 FFHQ 数据（左）和 CelebA-HQ 数据（右）生成的 1024×1024 分辨率图像 [191]。建议放大查看。经作者 [191, 第 8]页友好许可转载。

，这是通过自注意力实现的，因此其表示比其他方法所用的离散词元集合更加丰富。

按照这一方案，分别针对语音、图像和文本训练模型。对于图像，Data2vec 模型在受限训练集上以 86.2% 的 Top-1 准确率在 ImageNet-1k 上取得了新的最先进水平。对于语音数据，该模型在 LibriSpeech test-other 基准测试上的词错误率为 5.5%。对于语言处理，Data2vec 在 GLUE 上的平均得分为 82.9，优于 RoBERTa。这表明该模型能够有效处理多模态数据。可以预期，该模型将扩展为跨模态学习。

7.2.4 文本与图像联合处理

将 Transformer 应用于文本和图像后，联合处理这两种模态便成为显而易见的选择。为此需要三个步骤：

- 将图像和文本编码为能够保留其语义的嵌入；
- 设计强大的架构，以对这两种模态之间的交互进行建模；
- 开发有效的预训练任务。

学习到通用的视觉与语言特征后，这些预训练语言模型可以在各种下游视觉语言任务上进行微调。

对于文本-图像对的大规模数据集进行预训练时，需要使用$\boldsymbol{v}$、$\boldsymbol{u}$。我们假设有一个文本词元序列 $\boldsymbol{v}_1$、……、$\boldsymbol{v}_T$ 和一个图像特征或视觉词元序列 $\boldsymbol{u}_1$、……、$\boldsymbol{u}_R$，例如图像块。这样，我们就可以将两种模态的输入表示统一为嵌入序列。一个示例数据集是*COCO* 图像描述 [26]，其中包含 328k 张图像，涵盖自然上下文中的 91 种常见物体类型，并附有相应的图像描述（图 7.9）。其他数据集，如*Conceptual Captions*（CC）[153], *RedCaps* [34],和*Laion* [151]，分别包含 3.1M、12M 和 400M 张图像，并附有图像描述或描述性文本。



击球员击打棒球
裁判在一旁观看。

停车场上的一辆校车旁边有
积雪，旁边是一栋建筑。

两匹马拉着一辆干草车，车上有
两名男子。

图 7.9 MS-COCO 数据集 [26]：与数据集中示例图像相似的图像。相应的图像描述表明了详细程度。图像来源见表 A.3。

预训练任务的设计方式必须使模型能够根据剩余的上下文文本和图像特征，重建文本或图像的部分内容。对于跨模态掩码语言模型 （掩码语言模型），模型必须根据其他未被掩码的文本词元或视觉词元，预测被掩码的词元或图像块。在这里，可以采用不同的掩码策略，例如整词掩码、掩码文本片段，或对词元进行置换（第 3.1 节）。掩码区域预测 学习预测图像区域的内容。物体及其区域由人工或辅助模型进行标注。然后，模型需要预测该区域中的物体（或物体上的一个分布）。通过这种方式，模型学会定位图像中的物体。

CLIP [126, 127] 经过训练，可以预测一个得分，用于表明哪个图像描述对应哪个图像。给定一个由文本-图像词元对组成的批次（$\boldsymbol{v}_1$，$\boldsymbol{u}_1$）、……、（$\boldsymbol{v}_n$，$\boldsymbol{u}_n$），CLIP 必须预测该批次中所有 $n \times n$ 种可能的（$\boldsymbol{v}_i$，$\boldsymbol{u}_j$）配对中，哪些配对确实出现过。通过对比学习，CLIP 联合训练图像编码器和文本编码器，最大化批次中 n 个真实配对的图像嵌入与文本嵌入之间的余弦相似度，同时最小化错误配对的嵌入之间的余弦相似度，从而创建多模态嵌入空间。事实证明，这种使用正面和负面示例的对比训练优于其他替代方法。图像编码器采用了视觉Transformer（ViT），其图像块大小为 14×14 （第 7.2.2 节），其性能优于基于卷积神经网络的 ResNet [63] 编码器。文本前后分别添加了 [SOS] 和 [EOS]词元，并使用一个 12 层自回归 GPT 模型来计算嵌入。最高层中的 [EOS] 嵌入被用作整个文本的表示。

CLIP 使用*WIT* 数据 [127] 中的 4 亿个图文对进行训练，以便将图像与最匹配的图像描述关联起来。此外，下一个词元的预测被用作 GPT模型的辅助损失项。该模型可用于检索最适合某张图像的文本，或检索与某段文本最匹配的图像。

由此得到的模型获得了关于文本和图像的全面知识。其 Top-1 分类准确率达到 76.2 %，甚至超过了原始 ResNet50 在 ImageNet 上零样本 Top-1 分类的 75.0% 准确率。

无需使用 ResNet50 训练时所用的 128 万个训练样本，即可进行分类。因此，CLIP 可被视为一种“零样本分类器”。这一点在其他 27 个图像分类基准中的 16 个上也成立。在 CLIP 的特征之上拟合线性分类器，可使 CLIP 在 ImageNet 测试集上的准确率提高近 10%（[126]）。如果图像分布发生变化，例如变为素描，基于 CLIP 的分类器会更加稳健。零样本 CLIP 分类器大幅提升了有效鲁棒性，尤其是在应对分布偏移方面。这表明，将图像说明文本纳入视觉模型能够提升其性能和鲁棒性。

BriVL [46] 是一个面向中文的类似模型，它使用存储在队列中的更大规模负例集。它使用包含6.5亿个弱相关文本-图像对的庞大训练集，例如，一张生日蛋糕的图像，其图像描述可能是“生日快乐！许个愿吧”。它在跨模态检索和视觉问答方面取得了最先进水平的结果。

ALIGN [77]同样为文本和图像分别使用独立的编码器，并在顶层采用余弦相似度组合函数。图像编码器使用的是EfficientNet卷积神经网络。BERT经过训练，可为 [CLS]词元生成文本嵌入。同样，对于真实的图文对，模型会最小化相似度；对于随机配对，则最大化相似度。ALIGN拥有6.75亿个参数，并使用包含18亿个噪声图像对的庞大训练集。尽管数据存在噪声，该模型在 ImageNet Top-1分类上的准确率（85.5）仍略高于CLIP。

7.2.5 文本描述图像

自动生成图像的自然语言描述也称为图像标注 或图像描述。这项任务颇具挑战，因为它需要视觉感知、识别和现实世界知识，还需要将语言表达落地到图像空间中。符号落地 描述了词语如何获得意义，例如通过将一个词语与图像中的某个物体关联起来。除了确定并提取图像中的重要物体和详细信息外，模型还必须推断物体与场景之间的语义关系（图 7.9）。

当前用于描述图像的顶尖模型分为两个阶段：

- 预训练一个目标检测模型，将图像及其中的视觉对象编码为特征向量，
- 预训练一个跨模态预训练语言模型，使其能够关联文本和视觉特征，并为图像生成图像描述。

与语言翻译类似，人们使用各种指标来评估生成文本，例如 BLEU 或 ROUGE（第 2.3.3 节）。 [67, 112, 159]提供了图像描述技术的综述。

VilBERT [100] 旨在学习能够联合建模图像和自然语言的表示。它使用预训练的目标检测网络（Faster R-CNN [137]）提取边界框及其视觉特征。这些图像区域



图 7.10 标准边界框对象描述（左）以及可由 VinVL 生成的详细标注（右），其中包含视觉概念和属性信息 [193]。图像来源见表 A.3。

特征以及文本分别输入两个独立的Transformer编码器（双流架构）。随后，应用双向交叉注意力Transformer层来学习跨模态关系。VilBERT在Conceptual Captions数据上进行了预训练。

该模型针对不同任务进行了微调和评估。视觉问答 （*VQA*）用于回答有关图像的自然语言问题。VQA 被视为一个多标签分类任务，共有 3,129 种可能的答案。文本部分和图像部分的最终嵌入被输入分类器，以估计类别概率。在 COCO 测试集上，VilBERT 以 70.9% 的准确率创下新的最先进水平。基于图像描述的图像检索 是指根据描述图像内容的图像描述，从一组候选图像中识别出目标图像。该模型在 Flickr 数据集上进行了微调，召回率@1 达到 58.2%，从而创下新的最先进水平。

OSCAR [95]采用一种将图像中的相关物体与图像说明文本中的对应短语相连接的方法。作者使用自注意力学习这些对齐关系，而图像中检测到的额外对象标签可以作为参考点，显著提升对齐效果。Oscar 将每个输入图像-文本对表示为一个词语-标签-图像三元组（w； q； v），其中， w 是图像说明文本中的词语序列， q 包含图像中检测到的文本对象标签中的词语，而 v 是对应图像区域的集合。卷积神经网络模型（Faster R-CNN [137]）用于发现 q 中的物体以及对应的区域 v。为了预训练 Transformer 编码器，会对（w； q； v）中的部分词元进行掩码，然后模型学习预测这些掩码词元。此外，有时还会随机更改 q词语。模型还需执行一项额外任务，即识别这些修改。研究人员使用长度分别为 768 和 1,024 的序列训练了小型和大型模型，训练数据为包含 650 万个文本-图像对的公开语料库。随后，根据序列到序列目标对模型进行微调，以生成图像描述。在 COCO 图像描述上，该模型在 BLEU-4 （ 41.7% ）、 METEOR 和 ROUGE-L 指标上，以及在其他多个图像描述基准上，均达到了新的最先进水平。



图 7.11 SimVLM 编码器-解码器模型以图像（上方）和文本（中间）为输入，并生成文本输出（下方） [171]。图像块由 ResNet 的前几层进行编码。经作者慷慨许可转载图像 [171, 第 3] 页。

VinVL [193] 在包含 250 万张图像的三个图文语料库上进行预训练，并能够利用更丰富的视觉物体和概念集合生成视觉特征。VinVL 基于以卷积神经网络为基础的 ResNeXt-152 C4 架构 [179]，预训练了一个大规模物体属性检测模型。该模型并非只用一个名词描述物体，而是使用大量属性和详细信息，从而提升了其在图像语言联合任务中的性能（图 7.10）。该方法与 OSCAR 结合后，在图像描述方面取得了更高的最先进水平。**VIVO** [70] 是一种类似的 Transformer 模型，经过训练可使用 6,400 种不同的对象标签标注图像区域。VIVO 使用 COCO图像描述对进行微调，并学习生成描述句子，同时还使用图像描述数据中未出现的对象标签。由于 VIVO 可以利用大量成对的图像标签数据来学习丰富的图像描述，因此能够实现这一点。在测试集上，根据*CIDEr* 指标 [69]，，VIVO 生成的图像描述优于人工描述；该指标根据 tf-idf 对生成文本和参考文本 [169]中的共同词语进行加权计数。

SimVLM [171] 是一种 Transformer 编码器-解码器模型，使用 ResNet 的前三个模块从图像中提取上下文化图像块，并将图像词元与文本词元关联起来。随后，解码器预测文本序列的续写，如图 7.11 所示。该模型使用 18 亿个含噪图文对和 800 GB 文本文件进行训练。SimVLM 在*VQA v2* 基准 [56] 上的视觉问答任务中取得了新的最先进水平，准确率达到 80.3%。此外，在 COCO 图像描述上，该模型在视觉蕴含、视觉推理和图像描述任务中，按照 METEOR 指标（33.7）衡量，均达到了最先进水平。

Frozen 是一种经过训练、能够将文本与图像关联起来的基础模型。通过少样本学习，可以指示它回答有关图像的问题 [166]。该语言模型是一个经过预训练的自回归模型，包含 7B 个参数，使用包含 807GB 文本的 C4数据集进行训练 [129]。视觉编码器基于 NF-ResNet-50 [16]，并提供一个用于表征图像的嵌入向量。在训练过程中，图像嵌入会作为前缀，置于生成文本的词元嵌入之前。使用 *conceptualcaptions* 数据集训练视觉编码器，同时

冻结语言模型。训练目标是为图像生成图像描述。

推理过程中，会将若干由图像嵌入和词元嵌入组成的示例输入语言模型，由模型生成回答。例如，给显微镜配上图像描述“这是 *Zacharias Janssen* 发明的。”，给灯泡配上“这是托马斯·爱迪生发明的。”。经过五个种子样本，并输入一架飞机以及“这是由”后，模型会生成输出“莱特兄弟”。这样就可以即时定义不同的图像分类。这些样本展示了模型能够生成开放式输出，同时适应图像和文本，并利用其在纯语言预训练期间学到的事实。该模型是一个概念验证，展示了为图文任务生成少样本模型的一种方式。

7.2.6 从文本生成图像

通过在文本-图像对上进行训练，Transformer 可以获得生成与文本描述相对应图像的能力。借助语言模型依次生成下一个词元，可以预测视觉词元，随后将其合成为图像。不过，还有其他图像生成技术。

- 变分自编码器 (*VAE*)将输入图像压缩为较小的潜在表示，并尽可能好地重建图像。额外的损失项确保潜在表示的分布跟随高斯分布[79]。

- 生成对抗网络 (*GAN*)使用生成器将噪声向量 $\boldsymbol{s}$ 转换为图像 $\tilde{\boldsymbol{x}} = G(\boldsymbol{s})$。然后，判别器 $D(\boldsymbol{x})$ 负责将其输入分类为合成图像 $\tilde{\boldsymbol{x}}$ 或真实图像 $\boldsymbol{x}$ [53]。两个网络通过对抗损失交替训练。

Lee 等人 [91] 综述了文本驱动的图像生成与操纵技术。

目前有多种方法可以衡量生成图像的质量。*Inception* 分数 (*IS*) [150] 将基于卷积神经网络的 *Inception* 模型 [162]应用于每张在 ImageNet 上训练的生成图像，以获得条件类别标签分布；该分布应集中于少数类别，即熵较低。此外，测试数据还应生成许多不同的类别，这一点由定义的 IS 度量捕捉。*Fréchet Inception* 距离 (*FID*) [64] 是一种改进的度量，使用 ImageNet 分类器分布之间的 Fréchet 距离来衡量这些分布的相似度，同时考虑图中各得分的位置和顺序。*CLIP* 相似度分数 (CLIPSIM) [72] 基于 CLIP 模型（第 7.2.4 节）。它使用 CLIP 生成图像嵌入和文本嵌入，并计算二者的余弦相似度。

DALL-E [133] 使用一个含有 120 亿个参数的 *GPT-3* 自回归语言模型，根据文本描述生成新图像。图像的图像说明文本



图 7.12 根据一段自然语言描述（上），DALL-E [133]生成了多张图像。中间一行展示了 DALL-E 根据该描述生成的图像。下方一行展示了从 512 个样本中通过质量评分自动选出的最佳图像。经作者 [133, p. 6] 慷慨许可转载。

被字节对编码为 256 个词元。随后，每个 256×256 图像都使用离散变分自编码器压缩为由图像词元构成的32×32 网格。每个图像词元用 8,192 个可能值表示其 8×8 像素。描述词元与 $32 \times 32 = 1024$ 图像词元串联起来，形成 GPT-3 的输入序列。

在第一阶段，通过训练图像词元得到连续图像值。随后，使用 *Gumbel-Softmax*松弛 [75]（第 7.1.3 节）进行训练，以获得离散图像词元。在第二阶段，训练一个具有 64 个自注意力层和 12B 个参数的 *SparseTransformer* [27]，以顺序生成联合输入序列。对于图像词元，使用了特殊的注意力掩码：行、列或卷积注意力掩码。该模型在来自互联网的 2.5 亿个文本-图像对上进行训练。

对于图像生成，作者使用预训练的对比模型对从 Transformer 中采样的样本重新排序；该模型根据图像与图像描述的匹配程度分配得分。图 7.12 展示了使用该算法采样出的不同图像。与先前的 DF-GAN [165],模型相比，DALL-E 生成的图像被认为最逼真，并且在超过 90% 的情况下与图像描述更加匹配。同样，X-LXMERT [28] 生成的图像表现较差。

GauGAN2 [122, 149] 将分割映射、图像修复和文本到图像生成整合到一个模型中。它是首批能够为室内、室外、风景和街景等多种场景生成逼真图像的语义图像合成模型之一。最新版本还可以生成图像



图 7.13 由 CogView 根据文本输入（上方）控制生成的图像。输入文本可以影响图像风格。从 60 张样本图像中选出最佳图像。经作者许可转载 [40, 第 1] 页。

并根据文本输入生成图像。GauGAN2背后的模型使用了1000万张高质量风景图像进行训练。该模型的详细信息尚不清楚。

XMC-GAN [192] 是一种基于生成对抗网络的文本到图像生成模型，包含一个用于合成图像的生成器，以及一个经过训练、用于区分真实图像和生成图像的判别器。它最大化以下对应配对之间的互信息：（1）描述场景的句子与图像（真实或生成）；（2）具有相同描述的生成图像与真实图像；（3）图像（真实或生成）的区域与相关联的词语或短语。其目标是使匹配的配对（文本到图像配对以及真实图像到生成图像配对）具有较高的相似度分数，而不匹配的配对具有较低的分数。

对于输入文本，模型通过预训练BERT模块计算全局句子嵌入嵌入$_s$ 和词嵌入嵌入$_w$。嵌入$_s$ 与来自标准高斯分布的随机噪声 z拼接，形成全局条件，并将其传入多个上采样块，以生成 16×16 特征图。全局条件也用作条件批归一化层中计算缩放参数和偏移参数的条件。词嵌入嵌入$_w$ 作为“注意力自调制层”的输入，用于生成细粒度图像区域。在 MS-COCO 上，XMC-GAN 将最先进水平的 FID 分数（第 7.2.6 节）从 24.7 提升至 9.3，并且明显更受人工评估者青睐。同样，人工评估者在 77% 的情况下更喜欢 XMC-GAN 生成图像的图像质量；与另外三种最先进水平的方法（CP-GAN、SD-GAN 和 OP-GAN）相比，74% 的评估者更认可其图文对齐效果。

CogView [40]采用向量量化变分自编码器 （*VQ-VAE*）。在第一阶段，使用离散自编码器将图像转换为离散词元序列。在第二阶段，GPT 模型学习根据由 SentencePiece 文本词元组成的提示生成图像词元。为了生成图像词元，编码器将图像 $\boldsymbol{x} \in \mathbb{R}^{H \times W \times 3}$ 映射为$\boldsymbol{h} \times w$ 个图像块，并将其量化为可学习嵌入向量集合 $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\}$ $\boldsymbol{u}_i \in \mathbb{R}^q$[113]中的最近嵌入。解码器将嵌入映射回图像，并通过选择嵌入来尽可能缩小输出图像与输入图像之间的差异。

CogView 的 GPT 模型有 48 层，隐藏层维度为 2560，包含 40 个注意力头和 4B 个参数。模型的输入格式为“[*ROI*1] <文本词元> [*BASE*] [*BOI*1] <图像词元> [*EOI*1]”，并包含特殊词元。预训练任务是针对英语和中文的 3000 万个文本-图像对，从左到右预测词元。模型采用了与 BigBird 类似的稀疏注意力模式（第 3.2.1 节）。

如图 7.13 所示，CogView 在图像生成方面的性能与 DALL-E 相近。在模糊的 MS COCO 数据集上，它取得了最先进的 FID 分数，优于先前基于生成对抗网络的模型和 DALL-E，尽管 DALL-E 的参数量是它的三倍。由人类评估时，CogView 大幅胜过基于生成对抗网络的模型。然而，使用 CogView 生成图像相当缓慢，因为每张图像都是逐词元生成的。此外，量化会使图像出现一定程度的模糊。

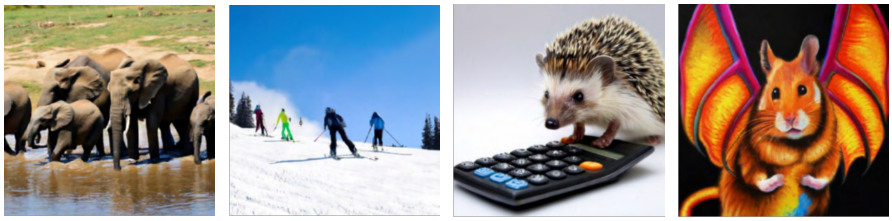
LAFITE[200] 是一种从文本生成图像的模型。图像生成基于*StyleGAN 2* [82], ，它通过调制卷积核的权重来创建各种图像属性 [177]。LAFITE 根据语言输入生成这些调制信号。它依赖预训练 CLIP 模型（第 7.2.4 节）的多模态语义空间，从文本 $\boldsymbol{x}$生成图像嵌入嵌入$(\boldsymbol{x})$，因此不需要额外的文本数据。随后，通过生成对抗网络架构，将该图像嵌入输入到一个类似 StyleGAN2 的图像生成模型中。在 MS-COCO 基准测试上，LAFITE 的零样本 FID 值为 26.9，优于 DALL-E（27.5）和 CogView（27.1）。在 MS-COCO 数据集上微调后，LAFITE 的 FID 分数为 8.1，优于 XMC-GAN（9.3）和其他生成对抗网络模型。请注意，LAFITE 只有 7500 万个可训练参数。

7.2.7 扩散模型修复被噪声破坏的图像

GLIDE [109] 是一种基于扩散模型的图像生成技术。扩散模型描述了通过迭代的前向扩散过程，系统且缓慢地破坏数据分布中结构的过程，例如添加噪声 [157]。对于数据 $\boldsymbol{x}^{[0]}$，例如像素值矩阵，我们可以应用高斯扩散分布 $q(\boldsymbol{x}^{[t]}|\boldsymbol{x}^{[t-1]})$，其中会添加一个期望值为$\mathbf{0}$、协方差为 $\beta\boldsymbol{I}$ 的高斯分布。这会得到一个序列 $\boldsymbol{x}^{[0]}, \ldots, \boldsymbol{x}^{[T]}$，其中最终分布 $\boldsymbol{x}^{[T]}$ 近似为协方差矩阵为单位矩阵的高斯分布（对二项分布也有类似结果）。

现在可以定义扩散过程的逆过程，即具有 $\boldsymbol{x}^{[t-1]} \sim p(\boldsymbol{x}^{[t-1]}|\boldsymbol{x}^{[t]})$ 的生成分布。Feller [47] 证明，当步长 β 很小时，条件分布 $p(\boldsymbol{x}^{[t-1]}|\boldsymbol{x}^{[t]})$ 近似服从高斯分布。因此，链 $\boldsymbol{x}^{[T]}$、.....、$\boldsymbol{x}^{[0]}$ 可以通过高斯分布生成

$$\boldsymbol{x}^{[t-1]} \sim N(\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{x}^{[t]}); \boldsymbol{S}_{\boldsymbol{w}}(\boldsymbol{x}^{[t]})) \quad \text{and} \quad \boldsymbol{x}^{[T]} \sim N(\mathbf{0}; \boldsymbol{I})). \tag{7.3}$$



一群大象
在泥水中行走

滑雪者群体正在
准备滑雪

一只刺猬在使用
计算器

一幅高质量的油画
一只迷幻仓鼠的
龙

图 7.14 GLIDE 根据下排的图像描述生成的图像。展示了 60 个样本中最佳的一个。经作者慷慨许可转载图像 [109, 第 7] 页。

该高斯分布完全由 $\boldsymbol{x}^{[t]}$ 的均值和协方差确定。

在训练过程中，通过 $q(\boldsymbol{x}^{[t]}|\boldsymbol{x}^{[t-1]})$ 从观测到的 $\boldsymbol{x}^{[0]}$ 开始生成噪声样本 $\boldsymbol{x}^{[t]}$。由此，可以通过在负对数似然 [65] 上优化变分下界 来重构逆向 $p(\boldsymbol{x}^{[t-1]}|\boldsymbol{x}^{[t]})$。使用训练好的模型，可以从样本 $\boldsymbol{x}^{[T]} \sim N(\mathbf{0}, \boldsymbol{I})$ 开始，并在一系列步骤 $\boldsymbol{x}^{[T-1]}$、.....、$\boldsymbol{x}^{[0]}$ 中逐渐减少噪声，其中

$$\boldsymbol{x}^{[t-1]} \sim p(\boldsymbol{x}^{[t-1]}|\boldsymbol{x}^{[t]}) \approx N(\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{x}^{[t]}); \boldsymbol{S}_{\boldsymbol{w}}(\boldsymbol{x}^{[t]})). \tag{7.4}$$

分布 $p(\boldsymbol{x}^{[t-1]}|\boldsymbol{x}^{[t]})$ 可以以图像类别为条件进行估计[37]。甚至可以用图像说明文本作为条件，而不是使用有限数量的图像类别。文本首先被编码为由 k 词元组成的序列，然后输入 Transformer 模型。Transformer 会输出一个类别嵌入以及 k 词元嵌入，并将它们用作额外的模型输入。在这里，会估计用于重建的常规噪声项$\boldsymbol{\epsilon}_{\boldsymbol{w}}(\boldsymbol{x}^{[t]}|\emptyset)$，此外还会估计以图像说明文本为条件的噪声项 $\boldsymbol{\epsilon}_{\boldsymbol{w}}(\boldsymbol{x}^{[t]}|c)c$。在进行无分类器重建 时，这两项会混合。

扩散模型由一个具有 23 亿个参数的 *U-Net* 模型近似表示 [144]，该模型会将 64 像素图像下采样到分辨率更低、特征更多的图像，然后再进行上采样。另一个具有 15 亿个参数的模型用于上采样至 256×256 分辨率。图像说明文本由一个具有 12 亿个参数的 Transformer 模型处理，最终的词元嵌入用于替代类别嵌入。

在测试中，GLIDE生成的图像质量很高，具有逼真的反射、纹理和阴影。该模型还可以组合多个概念（例如龙、迷幻风格和仓鼠），并为这些概念添加颜色等属性。在包含256幅图像的 MS-COCO 基准上，DALL-E 的 FID值为28，LAFITE为26.9，而GLIDE为12.2。在人工评估中，人们也明显更偏好GLIDE的结果。图7.14展示了一些由GLIDE生成的图像。值得注意的是，GLIDE的参数远少于DALL-E。GLIDE还可根据文本提示修复掩码图像块，例如，

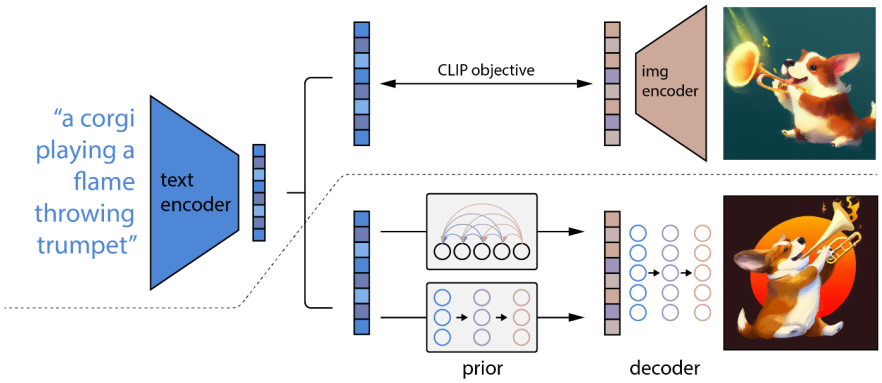


图 7.15 DALL-E 2[132]的高层概览。虚线以上展示了 CLIP 的训练过程：通过最小化图像嵌入与相应文本嵌入之间的差异进行训练。虚线以下展示了文本到图像生成过程：首先将 CLIP 文本嵌入输入自回归 Transformer（上方方框）或扩散先验（下方方框），生成图像嵌入。随后将该嵌入作为扩散解码器的输入，以生成最终图像。经作者 [132, p. 3]. 友好许可转载图像。

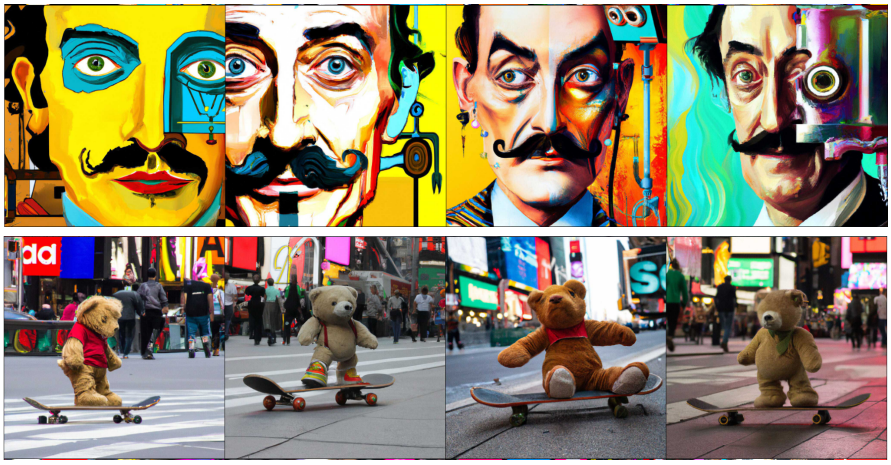


图 7.16 DALL-E 2 [132]根据提示"萨尔瓦多·达利的鲜艳肖像画，半张脸为机器人面孔"（上排）以及"时代广场上一只玩具熊踩着滑板"生成的随机样本。经作者 [132, p. 25, 27]. 友好许可转载图像。

"黑黄条纹的领带"。在大多数情况下，GLIDE的结果优于竞争模型，并且相应的图像块会以逼真的光照、阴影和纹理得到修复。最后，GLIDE可以为图像添加阴影和反射，并将简单的线稿转化为照片级真实感图像。

DALL-E 2 [132] 是 DALL-E 的改进版本，能够根据自然语言中的描述性句子创作更加逼真的艺术作品和图像。它分为两个步骤（图 7.15）：首先，根据先验 $p(z_i|y)$，生成基于文本描述y 的 CLIP（第 7.2.4 节）图像嵌入 z_i。然后，基于扩散的解码器根据图像嵌入 z_i生成图像 x。该解码器 $p(x|z_i, y)$对 CLIP 图像编码器进行逆向处理，具有非确定性，并且可以生成与给定图像嵌入对应的多个图像。在先验和解码器的训练过程中，CLIP 模型保持 Frozen。通过主成分分析，在保留几乎全部信息的同时，将图像嵌入 z_i 的维度从 1,024 降低至 319。319 个维度中的每一个都被量化为 1024 个离散桶。对于编码器，研究人员针对先验分别使用自回归模型和扩散模型进行实验。结果表明，扩散模型在计算上更加高效，并且能够生成质量更高的样本。示例见图 7.16。

解码器以图像表示为条件，可以生成保留图像语义和风格的多种变体，同时改变图像嵌入中缺失的非关键细节。CLIP 共享的嵌入空间支持以零样本方式进行语言引导的图像操作和修改。例如，可以将两张图像 x_1 和 x_2 融合，对 CLIP 嵌入空间中介于两者之间的所有概念进行插值。就 MSCOCO 零样本 FID 而言，DALL-E 2 的零样本 FID 为 10.4，优于 GLIDE（12.2）。人工比较表明，DALL-E 2 和 GLIDE 在照片真实感和图注相似度方面表现相近，而 DALL-E 2 生成的图像多样性更高。对于要求将两个独立物体（立方体）分别与两个独立属性（颜色）关联起来的提示，DALL-E 2 比 GLIDE 更难应对。现在，用户可以公开获取 DALL-E 来创建图像[115]。

Imagen [148] 是 Google 提出的文本到图像模型。它通过一个具有 46 亿个 Frozen 参数的预训练 T5-XXL 编码器-解码器 Transformer，将输入文本编码为文本嵌入。条件文本到图像扩散模型（7.3）将文本嵌入映射为 64×64 图像。随后，两个分别具有 6 亿和 4 亿个参数的超分辨率扩散模型分两步对这些小图像进行上采样，得到 256×256 和 1024×1024（图 7.17）。这些模型在 8.6 亿个图像-文本对上进行训练。

Nichol 等人 [110] 提出了对去噪扩散概率模型的一些修改，使其之后能够更快地进行采样，并在对样本质量影响很小的情况下实现更好的对数似然。它们具有与生成对抗网络相同的样本质量，但通过召回率衡量时实现了更好的模式覆盖。Imagen 也采用了该模型进行文本到图像转换，并将池化嵌入向量作为输入。该网络用于上采样，并经过扩展以提高内存效率、推理时间和收敛速度。图 7.18 展示了 Imagen 根据两个不同的图像描述输入生成的随机选取图像。

Imagen 在 *COCO*上实现了最先进的零样本 FID（Fréchet Inception 距离），得分为 7.3，优于 DALL-E 2，甚至胜过其他使用 COCO 训练的模型（表 7.2）。人工评估者根据照片真实感和与文本描述的对齐程度对 Imagen 进行了评估。在照片真实感方面，人类更倾向于

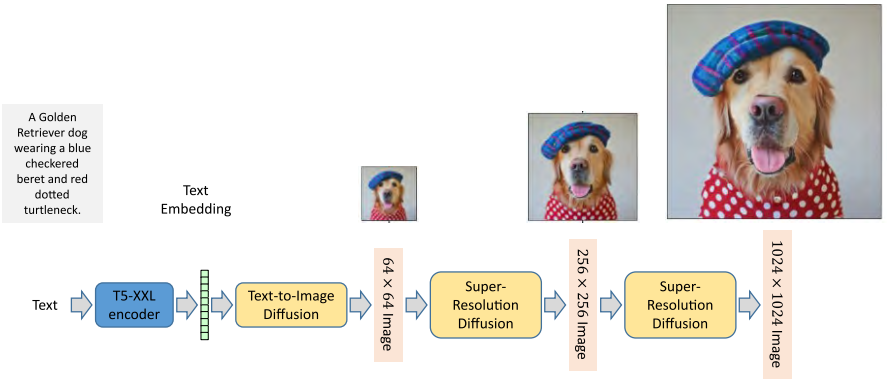


图 7.17 Imagen 通过预训练的 T5-XXL 文本编码器对输入文本进行编码。生成的文本嵌入由扩散模型转换为 64×64 图像 [148]。随后，两个超分辨率扩散模型将该图像提升至 1024×1024 分辨率。经作者友好许可转载 [148, 第 19]页。

在 39.5% 的情况下选择 Imagen 图像而非原始图像，这表明其真实感相对较高。在图注相似度方面，Imagen 的得分与原始参考图像不相上下。在 *DrawBench* [147]中，超过 60% 的情况下，评估者更偏好 Imagen 生成的图像，而非 DALL-E 2、GLIDE、VQGAN+CLIP 或潜在扩散生成的图像。作者强调，未来他们将增大语言模型的规模，因为相比增大扩散模型的规模，这有望带来更大的提升。他们没有发布 Imagen 的代码，也没有提供演示 API，因为这可能会被滥用，例如用于生成虚假图像。Gafni 等人 [48] 展示了如何扩展系统，以便在图像创作过程中为艺术家提供支撑。

Stable Diffusion 是另一种用于生成图像的模型，目前拥有 5.7B 个参数，可通过扩散生成最高达 1024×1024 像素的图像。示例如图 7.18 所示。它的工作方式与 DALL-E 2 类似，使用去噪 U-Net 对图像进行压缩和扩展 [142]。训练 Stable Diffusion 时，使用了来自免费开放的 LAION-5B 数据库 [12],的图像数据集，其中包含约 58.5 亿个经过 CLIP 筛选的图文对，是其前代 LAION-400M 的 14 倍。一个以 ImageNet 类别为条件的模型在图像生成任务中取得了 3.6 的 FID。该模型的一个变体会通过 CLIP 模型进行图像搜索，从每个训练样本的邻域中检索视觉特征相似的图像 [15]。模型在生成图像时会纳入检索到的图像。它可用于无条件图像合成、图像修复和随机超分辨率，并在显著降低计算成本的同时取得了有竞争力的性能。用于运行检索增强扩散模型的模型推理代码和模型权重现已开放 [141]，可供下载。用户大量使用该模型，每天生成 170 万张图像。



图 7.18 Imagen [148, p.6]（左）和 Stable Diffusion [142]（右）根据两种不同的文本描述生成的图像。经作者 [148, p. 6] 和[158],许可转载，相关致谢见表 A.3。

7.2.8 多用途模型

OFA（One For All）[170] 为一系列多模态任务提供了统一模型。它能够以文本词元和视觉词元的形式处理文本和图像。OFA 采用编码器-解码器 Transformer架构（第 2.3.1 节），并在各种文本和图像数据集上进行预训练。与 T5模型（第 3.1.3 节）类似，它会接收文本说明和图像，并生成相应的输出。

不同模态在同一空间中表示，文本、图像和宾语被离散化为统一输出词汇表。一个包含 256×256 像素的图像表示为 16×16 个图像块。每个包含 16×16 像素的图像块都被“词元化”为离散的视觉词元，使每个视觉词元与相应的图像块 [11]高度相关。此外，物体具有由标签及其边界框组成的特定表示。边界框的连续角坐标被均匀离散化为整数，作为位置标记（x_1； y_1； x_2； y_2）。最后，所有语言和视觉词元都使用统一词汇表，包括子词、图像编码和位置标记。

与 T5（第 3.1.3 节）类似，Transformer编码器-解码器由说明控制。它接收文本说明和输入图像，并生成相应的输出，即文本响应和图像。图 7.19 所示的示例描述了许多任务。通常，OFA 模型会在特定数据集上进行微调，以解决这些任务。

OFA 模型有一个 OFA_{Base} 变体，包含 6 个编码器层和解码器层、隐藏层维度为 768、12 个注意力头。$\text{OFA}_{\text{Large}}$ 变体包含 12 个编码器层和解码器层、隐藏层维度为 1,024、16 个注意力头以及 4.72 亿个参数。

在预训练期间，模型必须完成相应指令所要求的三项任务（图 7.19）。第一项任务是图像填充，模型需要重建图像的中心部分。这要求模型学习图像各部分之间的关系以及图像生成。第二项任务是目标检测，该任务建立图像各部分与语言描述之间的对应关系。最后一项预训练任务是文本填充，用于学习语言的结构。模型使用公开的数据集进行预训练，以完成不同任务；这些数据集包含超过 5000 万张图像和超过 160 GB 的文本。

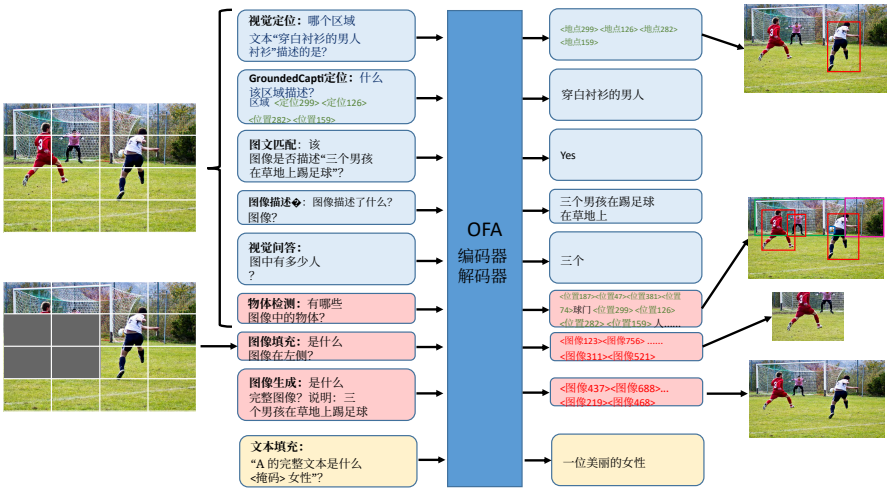


图 7.19 OFA [170, p. 3] 接收一条指令和一张输入图像，并生成文本以及（可选的）图像作为输出。图中展示了八条指令（左）各自对应的示例输出（右）。图像来源见表 A.3。

图像会调整大小至 384×384 像素，并采用固定的图像块尺寸 16×16 像素。每个图像块的特征向量由 ResNet 卷积神经网络的前三个模块计算得出。

针对图 7.19 所示的任务，模型在特定任务的数据集上进行微调，例如，使用 MS COCO 进行图像描述。除此之外，OFA 还在多项自然语言处理任务上进行微调，例如用于自然语言理解的 GLUE 基准测试、用于抽象式摘要生成的 Gigaword 基准，以及用于图像分类的 ImageNet-1K 数据集。推理时，作者采用束搜索，并开发了一种基于前缀树的搜索策略。这种基于字典树的搜索策略可确保 OFA 生成的输出局限于适当的候选集。

对于图像描述任务，模型在 MS COCO [26]上进行微调。凭借 43.5 的 BLEU-4 分数，该模型在 MS COCO 基准测试 [32]上创下了新的最先进水平。对于视觉问答任务，模型在 VQAv2 [56] 及类似数据集上进行微调。基于前缀树的搜索策略可确保 OFA 生成的输出局限于候选集。其准确率达到 80.0%，创下新的最先进水平。

对于视觉蕴含任务，模型必须判断图像与文本之间是蕴含、矛盾还是中性。OFA 在*SNLI-VE* [178] 上经过微调，在测试集上的准确率达到 90.2%，这一最先进的准确率比此前的最佳模型高 3.1%。指代表达理解要求模型根据语言查询定位图像区域。这里，模型在*Ref- COCO* 基准测试 [187] 及相关基准测试上经过微调，文本准确率达到 92.9%，大幅超越竞争对手，创下新的最先进水平。

对于图像生成，模型在 MS COCO [26]上经过微调。其 Fréchet Inception 距离（FID）为 10.5，优于 DALL-E[133] （27.5）和 GLIDE [109]（12.2）的得分；这两个模型的参数量分别为 12B 和 3.5B，远高于 OFA 的 472M。在排行榜上，只有 LAFITE（第 7.2.6 节）的 FID 值更低，为 8.1。请注意，竞争模型从 60 至 512 次试验的输出中选取结果，而 OFA 仅根据 FID 得分从 24 张图像中选出最佳结果。

在 ImageNet 图像分类任务中，OFA 不使用额外的标注训练数据，却取得了与 EfficientNet-B7（84.3%）相近的性能（Top-1 准确率为 84.9%），而当前最先进水平为 88.3%。令人惊讶的是，OFA 在仅使用语言的基准测试中也取得了良好结果，例如 GLUE 自然语言理解基准（第 4.1.1 节）和 Gigaword 上的摘要生成任务（第 6.4.1 节）。代码、演示和训练模型均可下载。

另一种多用途模型是 **NÜWA**，第 7.3.4 节对其进行了介绍。它能够根据文本实现逼真的文本到图像生成、图像编辑以及图像区域编辑。此外，NÜWA 还能够进行文本到视频生成，并预测下一个视频帧。

悟道2.0 [140, 143, 198] 是一个拥有 1,075B 参数的巨型混合专家模型，已在第 3.5.2 节中介绍。它基于 GLM 2.0 架构（第 3.1.3 节），结合了 BERT、GPT 和编码器-解码器 Transformer 的不同学习范式。在图像建模方面，它采用 CogView 方法（第 7.2.6 节）。但是，相关实现细节尚未公开。其训练数据包括 2.5TB 图像数据和 2.5TB 中英文文本数据（例如来自*Pile* 语料库 [49]）。悟道2.0 可应用于大量文本分析和生成任务，并在五项图像基准测试中达到或超过最先进水平，例如图像数据中的土地利用分类、图像生成和图形检索。

可用实现

- 视觉Transformer代码、训练模型和笔记本 github.com/google-research/vision_transformer
- OSCAR代码和预训练模型 github.com/microsoft/Oscar,
- VinVL代码和预训练Oscar-VinVL模型 github.com/pzzhang/VinVL。
- DALL-E代码和笔记本 github.com/openai/DALL-E
- OFA模型代码、预训练模型和在线演示 github.com/OFA-Sys/OFA
- GLIDE代码、训练模型和笔记本github.com/openai/glide-text2im
- Stable Diffusion https://github.com/CompVis/latent-diffusion

7.2.9 总结

近年来，视觉Transformer（ViT）已成为卷积神经网络（CNN）的有力替代方案，适用于图像识别任务。ViT模型

在多个基准测试中，ViT模型的准确率优于卷积神经网络，且所需的计算开销要低得多。

图像处理基础模型接收图像块作为输入。这些图像块的嵌入通过不同方法生成，例如对图像像素进行线性变换、使用卷积神经网络（CNN）模型的前几层、使用变分自编码器（VAE）或使用生成对抗网络（GAN）。扩散模型采用了完全不同的方法，通过添加噪声来逆转图像退化过程（GLIDE）。研究表明，将图像块的表示离散化有助于减少噪声和对低级纹理的依赖。

包含文本有两种方法。有时，文本词元和图像词元由不同的 Transformer 分别处理，随后最小化两类嵌入之间的距离（CLIP），或者通过交叉注意力使所得嵌入产生相关性（VilBERT）。另一种方法是将文本词元和图像词元连接起来，作为基础模型的输入（自编码器、自回归模型或编码器-解码器模型）。最近的模型（DALL-E、CogView、OFA）似乎更倾向于单流架构。预训练会采用多种不同的任务，包括掩码语言模型（MLM），其中需要重建被掩码的图像词元和语言词元；掩码区域分类（MRC）；以及掩码区域重建。句子-图像对齐（SIA）用于分类图像-文本对是否相互匹配。

图像描述生成会构建一个句子，对图像进行刻画（VilBERT、OSCAR、VinVL、SimVLM），使用流畅且正确的语言；根据人工评估，这通常是准确的描述。较长图像描述的生成尚未得到研究，可能与视频描述生成更为相关。已有研究用于探究视觉语言模型中的注意力模式 [19]。

与图像描述相匹配的图像创作在过去一年中实现了质量上的巨大飞跃。目前使用了多种架构：生成对抗网络（GAN）、扩散模型和VAE。这些模型通常与预训练语言模型结合使用。纯Transformer模型似乎具有优势（OFA），但DALL-E 2.0等扩散模型正在获得发展势头。通常会生成一组图像样本，并通过质量评分自动选择最佳图像。模型生成的图像分辨率通常为 256×256，并且已经涵盖许多详细信息。预计明年它们将达到更高的分辨率，例如 1024×1024。

Cao 等人 [19] 研究了视觉和语言模型的内在机制。他们得出结论：层数越深，多模态融合就越交织。通常，在做出决策时，文本模态比图像特征更具主导性，因为模型在推理过程中倾向于关注文本而非图像。研究发现，一部分注意力头专门用于跨模态交互。某些注意力模式能够将图像区域与文本中的词语对齐。最后，预训练的视觉和语言模型编码了丰富的语言学知识，因此其语言能力并未下降。

近年来，人们提出了多用途模型，这些模型经过训练，能够解决大量不同的语言、视觉和语言-视觉任务。一个示例是 OFA，它拥有 472M 个参数，远少于 DALL-E（12B）。OFA

是一种以图像和文本词元为输入的 Transformer 编码器-解码器，并通过类似 T5 的文本指令进行控制。它在图像描述、图像生成、视觉问答、视觉蕴含，甚至纯语言任务上都达到了最先进水平。与之形成对比的是拥有 1,750B 个参数的庞大悟道 2.0 模型，它基于采用混合专家架构的编码器-解码器 GLM 模型。该模型声称在多种图像和文本任务上达到最先进水平，但目前尚无已知的技术细节。

未来，这些文本图像模型有望扩展至视频、语音和三维等其他模态。此外，还将使用更多数据；同时，模型也会通过检索技术得到增强，以纳入更多外部且最新的知识。文本图像模型是迈向符号落地的重要一步，它能够将符号（词语）与其意义关联起来。

7.3 视频理解与生成

随着 Web 成为一种不断发展的交流手段，仅通过文本和图像表达内容往往并不足够。视频汇集了三种格外引人注目的元素：图像、运动和音频。因此，视频作为交流手段正变得越来越重要。YouTube 每月有 20 亿活跃用户，TikTok 则有超过 10 亿用户，平均每天使用 52 分钟。因此，视频的自动分析、理解和生成具有极高价值。对于视觉数据而言，视频提供了最全面的自监督信号。视频的各种模态，如图像、语音、自动语音识别文本和字幕，在时间上彼此对齐，无需人工标注。海量的多模态视频有望使基础模型习得关于视觉世界的模型。

7.3.1 视频处理基础

视频分析与理解比图像处理更具挑战性，因为视频还包含额外的时间维度，而且通常需要同时处理图像、语音以及语音转录文本或字幕。近年来，预训练语言模型已用于视频理解。与卷积神经网络和循环神经网络相比，Transformer 的主要优势在于能够同时捕获全局信息并进行并行计算。此外，Transformer 简洁且可堆叠的架构使其能够在更大数据集上进行训练。表 7.3 列出了使用基础模型处理视频的主要方法。

早期的图像处理模型，例如生成对抗网络或 Image Transformer，逐像素地分析图像。然而，由于计算和内存开销过高，这种方法已无法用于视频，因此必须聚合图像信息，

以适应 Transformer 有限的序列长度。为此，人们开发了专门的时空聚合模块。

- 一种简单的解决方案是聚合 30 个视频帧（VideoBERT）。
- 可以通过考察三维视频块来处理视频，这些视频块覆盖少量帧中的一个小像素区域。可以在不同的时间层级上聚合视频和文本，并计算各层级之间的关联（COOT、MTV）。区域聚合和时间聚合可以分开进行（CoVeR）。
- 此外，还可以处理视频块以提取显著信息。一个示例是使用变分自编码器进行视频量化（VQ-VAE）；变分自编码器此前已用于图像处理，例如 DALL-E 或 CogView（第 7.2.6 节）。图像块可以沿时间维度扩展，形成三维体素（VATT、Omnivore）。
- 视频可以划分为较短的时间片段。先前的片段可以参与自注意力计算，但无需更新先前的嵌入（MeMViT）。

为了进一步降低计算开销，可以采用稀疏自注意力，此时注意力主要集中在邻近的视频像素上。

无监督训练可以采用类似 BERT 的方式进行。例如，可以根据邻近的视频词元和文本词元预测掩码视频词元 [145]。同样，也可以根据邻近的文本词元和视频词元预测掩码文本词元。可以使用对比学习来区分真实的文本-视频对和随机配对。其他任务包括对视频与某段文本是否相互匹配进行分类、预测下一帧，或重建打乱的视频词元或文本词元的顺序。Islam 等人 [73] 、Khurana 等人 [85]和 Ruan 等人 [145]近期发表了关于视频理解的综述。

视频有多种训练数据来源。*Kinetics* [83]是一个大型、高质量数据集集合，包含 306k 个时长为 10 秒、聚焦于人类动作的视频片段。Kinetics 400、600 和 700 这些变体分别标注了 400、600 和 700 个类别。图 7.21 展示了标注视频的示例帧。*Moments in Time* [107] 是一个包含 800k 个带标签的 3 秒视频的集合，视频涵盖人物、动物、物体或自然现象，捕捉动态场景的要旨。*Epic-Kitchens-100* [33] 由 90k 个第一视角视频组成，总时长达 100 小时，均在厨房中录制。每个视频都标注了一个“名词”和一个“动词”。通常会报告三个准确率得分（“名词”“动词”和“动作”）。动作得分用于评估名词-动词配对是否正确，最为重要。*Something-SomethingV2* [55] 包含 220k 多个短视频片段，展示人类与日常物体的互动。不同类别的视频中会出现相似的物体和背景。与其他数据集不同，这些数据考验模型根据运动线索区分类别的能力。

表格 7.3 用于视频的预训练语言模型主要技术。括号中的数字表示参数数量

模型	方法	基准测试
视频转文本		
VideoBERT	将视频划分为 30 个片段并生成由卷积神经网络生成嵌入。对嵌入进行聚类，采用 *k*-means。自动语音识别语音生成文本词元。将输入拼接后送入 BERT。	YouCook II 视频图像描述 4.3 BLEU-4。
COOT	图像、视频和文本在 3 个不同的层级中进行处理。每个层级使用单独的 Transformer。每个层级都特别注重协作。层级。（10.6M）	YouCook II 视频图像描述 11.3 BLEU-4。
DeCEMBERT	视频二维和三维特征、区域描述、自动语音识别文本。输入线性变换为单个 BERT。	YouCook II 视频图像描述 11.9 BLEU-4。
VATT	生成图像-时间补丁，为每个视频、音频和文本分别使用 BERT。对比估计以缩小嵌入距离。	Kinetics-400 动作识别 81.1%
Omnivore	图像、视频和三维视图经过转换后被输入采用移位窗口的 Swin Transformer。	Kinetics-400 动作识别率为 84.1%（无额外数据）
MeMViT	结合过去视频记忆的注意力计算视频帧。记忆未经训练。使用记忆压缩模块进行池化。	动作识别准确率 EPIC-KITCHENS-100 48.4%
CoVeR	独立的图像与时间聚合。并行用于图像与视频识别的微调	Kinetics-400 动作识别 87.2%
MTV	通过多视图进行时间聚合。使用每个视图使用不同的视觉Transformer（1B）	Kinetics-400 动作识别率 89.1%
Merlot	联合处理视频和自动语音识别文本。用于文本和视频的掩码语言模型。重新排列打乱顺序的帧。	视觉问答准确率为43.1%
Flamingo	通过视觉Transformer处理图像和视频（80B）。通过语言模型（Chinchilla）、适配器和交叉注意力纳入图像信息层。支持少样本提示	在全部 8 项图像基准测试和全部 8 项视频基准
文本到视频		
视频 Transformer	将视频划分为具有不同维度的三维块，分布在不同层中（373M）	自回归视频生成 FVD 在 BAIR机器人上的得分为 94 data
NÜWA	图像、视频和文本数据表示为三维词元。由 VQ-GAN 离散化。使用局部注意力计算。针对文本到图像、视频预测和文本到视频进行训练。更多应用	自回归视频生成 FVD 分数为 86.9（BAIR机器人）数据（最先进水平）文本到视频 FID 图像指标 Kinetics 上为 28.5
Imagen Video	基础视频生成模型 +多个空间和时间视频超分辨率扩散模型	约 9.0 的 FVD 分数具有 56 亿个参数

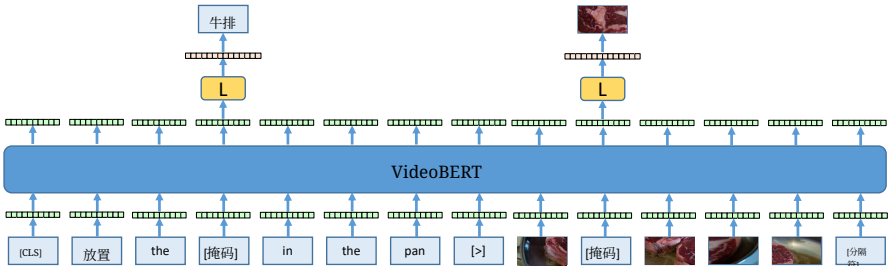


图 7.20 由自动语音识别生成的文本和相应的视频词元是 VideoBERT [160] 的输入。两种模态均由特殊词元界定。需要预测掩码词元。图像来源见表 A.3。

7.3.2 视频描述生成

视频描述生成旨在自动生成对视频的自然语言描述。视频描述生成比图像描述生成困难得多，因为视频中的时空信息以及相应的自动语音识别文本都会带来额外的复杂度。另一方面，YouTube 等平台上有大量视频可在互联网上获取，可用作训练材料。早期的模型（如 VideoBERT [160]）只是将 BERT 模型应用于由词元构成的文本和视频帧联合表示。 [124]提供了最新的综述。

VideoBERT [160] 将 BERT 模型应用于视频文本对。视频被划分为每段 30 帧（1.5 秒）的片段，并由带有时间卷积 [180],的 S3D 卷积神经网络处理，生成大小为 1,024 的片段嵌入向量。通过 k-均值聚类，将片段嵌入划分为 20,736 个簇，并量化为视频词元。语音经自动语音识别处理后被划分为句子。文本使用 WordPiece 分词，词汇表包含 30k 个词元。收集与句子时间段对应的视频词元，组成视频词元序列。如图 7.20 所示，视频词元被附加到相应的文本词元之后，两者之间以特殊词元分隔。需要注意的是，也可以仅使用文本或仅使用视频进行训练。

TheBERT$_{\text{LARGE}}$ 模型在包含 31.2 万个烹饪视频、总时长为 966 天的视频集上进行预训练。文本通过自动语音识别获得。训练任务包括掩码语言模型（MLM）、掩码帧建模（MFM）和视频语言建模（VLM）的掩码预测。VideoBERT 在 YouCook II 数据集的视频描述生成任务上达到最先进水平，BLEU-4 分数为 4.3。

COOT [51]通过嵌入向量，以通用表示共同处理图像、视频和文本信息。在视频表示中，时间作为第三维度加入图像的二维描述中。COOT 模型考虑了 3 个不同层级结构的数据：帧/词、片段/句子和视频/段落。每个层级都由一对 Transformer 处理输入。为建模层内协作，COOT 使用了一个特征聚合层

以聚焦于低级实体之间的时间交互。为将信息聚合到句子级，模型使用一种特殊的注意力公式，其中所有对应的嵌入都会参与标量积。另一个损失项旨在缩小句子与片段编码之间的差异。在顶层，上下文变换器将文本嵌入与视频嵌入连接起来。

该模型使用带有单个场景字幕和较长片段字幕的视频进行训练。之后，模型便可为新视频生成字幕。在 YouCook2 视频字幕基准数据集上，该模型可将最先进水平大幅提升至 BLEU-4 11.3 分。此外，该模型还可用于其他任务，例如输入文本描述或视频场景后进行搜索。由于该模型仅包含 10.6M 个参数，因此预计通过增大模型规模，性能可大幅提升。

DeCEMBERT [164] 旨在通过区域描述 增强视频，并补充语音识别提取的自动语音识别文本。输入文本以 BPE词元表示。视频的每一秒都由预训练的 ResNet-152卷积神经网络提取的二维特征 [63]以及由 3D ResNeXt卷积神经网络提取的运动特征 [179], 来表征，随后二者共同映射为嵌入向量。视频嵌入与语音识别文本表示拼接成单一序列，作为输入用于 12层自编码器的预训练和下游任务微调。为使视频与自动语音识别字幕对齐，模型采用约束注意力损失，促使模型从候选池中选出匹配度最高的自动语音识别字幕。在 120 万个 YouTube 教学视频上进行预训练时，模型通过掩码词元（掩码语言模型、视频语言建模）以及判断文本是否对应某个视频的分类任务，学习文本与视频之间的关联。在 YouCook2 图像描述任务上，该模型将最先进水平提升至 BLEU-4 11.9 分。此外，DeCEMBERT 在视频检索和视频问答方面也取得了良好结果。

7.3.3 视频动作识别

VATT [2] 以互联网视频的原始 RGB 帧、音频波形以及语音音频的自动语音识别文本作为输入数据。尺寸为 $T \times W \times H$ 、包含 T 帧的视频会被划分为由 $\lceil T/t \rceil * \lceil H/h \rceil * \lceil W/w \rceil$ 个图像块组成的序列，其中每个图像块都是 $\mathbb{R}^{t \times h \times w \times 3}$ 中的一个体素 ，并额外包含一个颜色维度。这是对 ViT 图像块的扩展。位置编码是以下各项之和：$\boldsymbol{e}_{i,j,k} = \boldsymbol{e}_{\text{temp};i} + \boldsymbol{e}_{\text{horiz};j} + \boldsymbol{e}_{\text{vert};k}$，其中每个加数都是长度为 d 的可学习向量。原始音频波形被划分为 t' 个片段，并为每个片段分配一个可学习的位置嵌入。对于文本，会创建一个词汇表，并将每个单词映射到一个可学习的嵌入。*DropToken* 过程会随机移除视频或音频词元，以降低计算成本并改善正则化。

VATT 将每种模态线性投影为长度为 d 的特征向量，并将其输入独立的 BERT 编码器。该模型使用噪声对比估计来缩短音频嵌入与视频嵌入投影之间的距离。正样本对取自视频中的相同位置，负样本对则取自



图 7.21 两个视频及其描述标注（左），类似于 Kinetics 数据集中的视频 [83]。图中展示了这些视频的代表性帧。显然，有时仅凭单帧不足以做出判断，例如无法区分“运球”和“扣篮”。图像来源见表 A.3。

不同位置。模型采用类似的标准来缩短视频嵌入与文本嵌入之间的距离。训练数据涵盖以 10 fps 从 *HowTo100M* 数据 [105] 中提取的 32 帧片段。最大的模型拥有 4.15 亿个参数。在 Kinetics-400 上进行动作识别时，该模型以 82.1% 的 Top-1 准确率和 95.6% 的 Top-5 准确率达到了最先进水平。

Omnivore [52] 是一种模型，可使用完全相同的模型参数对图像、视频和单视角 3D 数据进行分类。单视角 3D 数据是一张带有额外深度通道的彩色图像。图像、视频和单视角 3D 模态会转换为嵌入，并输入 Transformer 模型。图像被划分为图像块；视频被拆分为覆盖不同图像区域的时空管；单视角 3D 图像则被转换为 RGB 图像块和深度图像块。图像块通过线性层投影为嵌入。图像和视频的 RGB 图像块使用同一个线性层，而深度图像块则使用单独的层。空间维度和时间维度分别使用独立的位置嵌入。

Omnivore 采用 *Swin Transformer* （第7.2.3节）作为基础模型，这是一种使用移位窗口的层次化视觉Transformer。自注意力涉及来自空间和时间上邻近图像块的图像块嵌入。模型在 ImageNet-1K 数据集（用于图像分类，包含120万张图像）、Kinetics-400 数据集（用于动作识别，包含24万个视频）以及 *SUN RGB-D* 数据集 （ 5000 个样本，用于单视图三维场景分类）上进行联合训练，并通过特定于数据集的线性分类层转换最终嵌入。在不使用额外数据的情况下，Omnivore 在 Kinetics-400 上取得了84.1%的动作识别准确率，排名第二。经过微调的 Omnivore 在两个视频分类基准测试上达到了最先进水平。当使用 RGB 和三维通道时，Omnivore 在 NYU-v2 基准上再次取得了最先进水平。

MeMViT [173] 旨在处理时长超过 5 秒的视频，这与当前大多数模型不同。MeMViT 以在线方式处理视频，并在每次迭代中缓存 Transformer 的键向量和

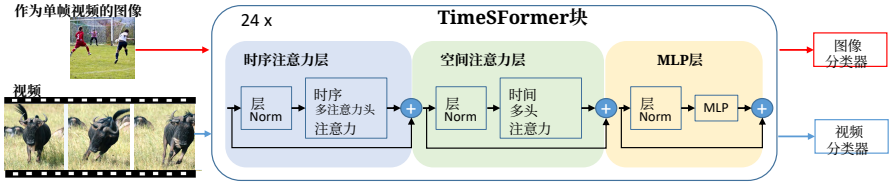


图 7.22 在微调过程中，CoVeR [190, p. 5] 同时在多个图像和视频数据集上进行训练。由于类别定义不同，每个数据集都有自己的分类器。图像是单帧视频。因此，图像分类不受时间注意力的影响。表 A.3 中提供了图像来源。

值向量作为内存。借助内存，该模型能够以极低的额外成本获取先前的上下文，用于长期建模，因为记忆嵌入无需训练。当前视频片段的查询会关注扩展后的键向量和值向量集合，这些向量同时来自当前时间和过去。类似于 WaveNet 的扩张卷积，[114], 更高层会关注更久远的过去，从而产生显著更长的感受野。此外，采用可学习池化的记忆压缩模块能够有效减少内存占用。

视频被拆分为一系列短$T\times H\times W$片段，并按顺序进行处理。与 MTV 类似，模型采用多种分辨率，从较小图像块的细粒度建模开始，到后续阶段对较大图像块进行高级建模，此时嵌入的维度会增加。阶段之间的聚合通过步幅池化完成。内存表示被冻结，不会因优化而改变。模型在 Kinetics-400 数据上进行预训练，见图 7.21。在 AVA v2.2 数据集上，MeMViT 达到 35.4% 的平均精度均值（mAP）。在动作预测数据集（EPIC-KITCHENS-100）上，其最先进水平为 17.7% 的前5项召回率。在 EPIC-KITCHENS-100 上进行动作识别时，MeMViT 的准确率达到 48.4%。

CoVeR [190] 评估不同预训练策略对分类准确率的影响。作者采用了一种特殊的 Transformer 架构，该架构在同一视频帧中相关区域之间使用空间注意力层，并在视频片段中相邻帧之间使用时间注意力层。CoVeR 首先在包含 30 亿张图像的*JFT-3B* 基准测试 [189] 上对模型进行预训练，这些图像标注了约 3 万个类别层级标签。预训练期间，所有时间注意力层都会被移除。微调期间，该模型同时使用多个动作识别和图像数据集（Kinetics 各版本、ImageNet、Moments in Time、SomethingSomethingv2）进行训练，包含 24 层，以构建稳健的视频数据空间和时间表征（图 7.22）。在 Kinetics-400 动作识别任务中，CoVeR 的准确率达到 87.2%；在 Moments in Time 动作分类任务中，其准确率达到最先进水平，为 46.1%。

MTV [185] 通过对输入视频的多种输入表示（视图）进行时间聚合。MTV从输入视频中提取多个时间

跨度的词元。由较长时间间隔衍生的视频词元能够捕捉整体场景描述，而从较短文本片段中提取的视频词元则能够捕捉细粒度的详细信息，例如人物的手势。不同的Transformer 编码器用于处理这些不同的视图，其中短文本片段模型具有更高的容量。

不同的编码器通过横向连接进行交互，以融合跨视图信息。在相邻视图之间计算跨视图注意力，其方式类似于Transformer中的多头交叉注意力（第2.3.1节）。请注意，这些融合运算仅针对特定层执行。来自所有视图的词元由全局编码器聚合，该编码器执行最终分类。

模型使用视觉Transformer权重（第7.2.2节）进行初始化，并使用帧数为32、分辨率为 224×224的视频进行训练。结果表明，跨视图注意力在融合不同视图的信息方面优于其他方法。此外，使用三个视图比使用更少的视图能够获得更好的结果。最大的模型拥有超过十亿个参数，在Kinetics-400上的动作识别达到了89.1%的最先进的准确率。

视听语音识别 [152] 将预训练语言模型应用于视听语音识别。通常，音频会以10毫秒为步长转换为80维对数Mel滤波器组特征。视频会裁剪至嘴部附近的区域，并转换为长度为512的视频嵌入。两种嵌入经过拼接后，输入具有17层的Conformer编码器（第7.1.2节）。该模型在LRS3-TED基准 [1] 上的唇读任务中超越了此前的最先进水平，词错误率为19.3%。如果使用两种模态，词错误率会降至1.6%。如果加入喃喃噪声，LRS3-TED上仅使用音频的自动语音识别词错误率会升至6.1%，而使用两种模态进行语音识别时，词错误率仅为2.9%。还有另一种关联视频和音频的方法，即生成与视频中运动的速度、情绪和节奏相匹配的视频背景音乐 [38] 。

Aloe [39]不仅希望简单描述图像或视频，还旨在解释场景或对场景进行推理。该模型使用无监督的宾语分割模块，将每幅图像划分为对象表示。Transformer接收问题和包含对象表示的图像描述。在多个视觉推理基准测试中，模型需要回答复杂问题，例如“为什么会发生某件事？”这类解释性问题、“接下来会发生什么？”这类预测性问题，以及“在未见过的情境下会发生什么，例如移除一个对象后会怎样？”这类反事实问题。该模型能够在几乎所有基准数据集上提升最先进水平。

Merlot [188] 是一种视觉语言模型，可从包含数千帧的视频及其自动语音识别文本中学习多模态世界表示。它使用图像编码器对每一帧进行编码，使用学习得到的嵌入表示词元，并由类似 RoBERTa 的 Transformer 联合处理这两种表示。第一项预训练任务使用对比损失，将语言转录嵌入与相应的视频嵌入进行匹配。掩码语言建模任务要求替换被掩码的语言词元。时间重排序任务则要求对打乱顺序的视频帧重新排序。因此，Merlot 不仅学习将图像与时间上对应的词语相匹配，还能从全局角度理解随时间发生的事情，从而获得时间常识知识。该模型使用 600 万个未标注的 YouTube 视频进行训练。

Merlot 在 12 项下游基准测试中优于最先进的方法，这些基准测试涵盖短视频和长视频。例如，在 MSRVTT-QA 上进行视觉问答时， [182]取得了 43.1% 的新最先进水平。另一种用于复杂事件抽取的相关模型 [93]采用了类似的对比学习方法。

Flamingo [3]是一种视觉语言模型，能够处理任意交错排列的图像、视频和文本数据序列。Flamingo采用参数量为70B的预训练语言模型*Chinchilla* ，该模型在大规模、多样化的文本语料库上训练而成（第3.1.2节）。语言模型的编码器块采用冻结的参数。借助这个子模型，Flamingo具备强大的生成式语言能力，并能获取存储在Chinchilla权重中的大量知识。与*Frozen* （第7.2.5节）类似，它可以通过少样本学习进行指令引导，从而回答有关图像的问题 [166]。

为处理图像和视频，研究者预训练了一种对比式图文方法（图 7.23）。作者使用了ResNet的一种变体 [16]。视觉编码器在我们的图文对数据集上采用对比目标进行预训练，并使用了 [127]中的双项对比损失。与CLIP（第7.2.4节）非常相似，图像编码器的均值池化输出与BERT模型的均值池化输出之间的点积用于计算相似度。该模型从图像中提取语义空间定向特征，包括颜色、形状、性质、物体位置等。这个模型单独进行预训练，并在Flamingo的主要训练期间冻结参数。

训练了两个模块，以便与这些冻结模型进行交互。第一个是感知器重采样器，它接收来自视觉编码器的时空特征，并输出一组固定大小的视觉词元（通常为 64 个）。无论输入图像的分辨率或输入视频帧的数量如何，它都会分别为单张图像和视频生成这一输出。随后，通过交错插入的交叉注意力层，将提取出的视觉词元纳入语言模型。这样，语言模型就能在每一层整合视觉信息。冻结的语言模型和视觉模型分别有 70B 和 435M 个参数，而可训练层有 10B 个参数，重采样器有 194M 个参数，总计 80.6B 个参数。

Flamingo 使用包含 182GB 文本的多个数据集进行训练。该数据集合还补充了更多混合文本、图像和视频序列，其中总计约有 2.3B 张图像和 27M 个视频。

如图 7.24 所示，Flamingo 只需预测图像-文本混合序列中的下一个文本词元，就能回答关于单个图像的问题。最简单的问题形式是询问场景中物体的描述，如右上方的示例所示。对场景进行解读则更加困难，因为语言模型需要借助世界知识来判断图像中哪些方面值得关注。在许多示例中，Flamingo 至少可以完成一步隐式推理。有些物体并未在提示中明确命名（例如大象），但问题会直接询问它们的性质。为了回答这些问题，模型需要推断所指的物体，然后召回相关知识以组织答案。这可能产生单一答案（如关于卡车上的大象的问题），也可能产生扩展对话，在其中模型可以回答一系列关于某个

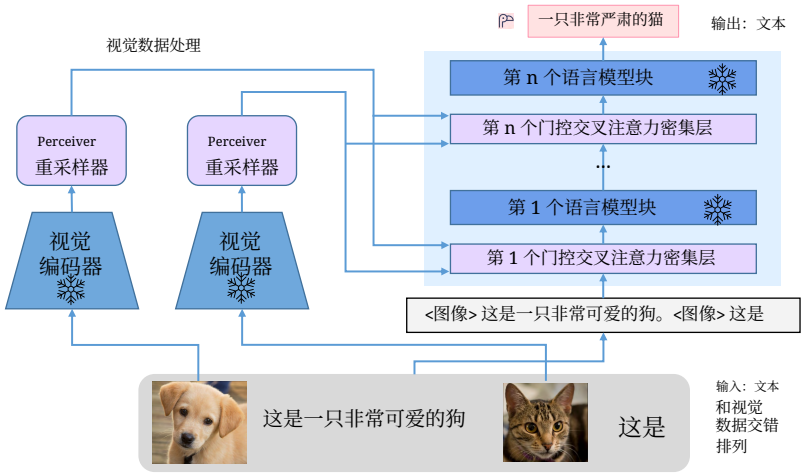


图 7.23 Flamingo [3] 接收的输入由图像、文本和视频按任意顺序组成的序列构成。图像和视频由类似 CLIP 的冻结视觉编码器处理。可训练感知器重采样器将它们缩减为有限数量的图像词元，再由可训练交叉注意力层将其纳入语言模型。语言模型生成的输出是输入序列的自然延续。图像经改编自 [3]，并获作者慷慨许可，鸣谢信息见表 A.3。

图像的查询（例如破坏沙发的狗）。即使经过多次交互，Flamingo 仍能成功关注图像，并回答需要解读图像的问题。作者观察到，模型可以分别关注多幅图像，也能正确处理简单的比较和推理。Flamingo 的对话能力可以让非专业终端用户无需微调即可获得答案。

同样，Flamingo 也能够回答有关视频的问题，如图7.25所示。不过，该任务的性能并不像期望的那样稳定。

Flamingo 能够对混合文本-视频-图像序列执行 少样本提示 。示例如图7.26所示。这里提供了若干图像，添加的文本通过示例明确了提取回答的所需方式。第一行要求从图像中提取文本，第二行则要求统计同一类型物体的数量。通过这种方式，可以动态地指导模型执行大量任务，例如图像描述、视觉对话、分类或视觉问答。

该模型在 9 个图像-文本基准测试上进行了性能测试，涵盖场景描述、视觉对话和视觉问答，其中包括 MS-COCO图像描述。在八个混合媒体基准测试中，Flamingo 仅用 16 或 32 个样本，就在所有基准测试上大幅刷新了少样本最先进水平。对于其中三个基准测试，这一结果甚至优于先前微调后的最先进水平。在 ImageNet Top-1 分类任务上，Flamingo 达到 76.0%，而微调后的最先进水平为 91.0%。视频测试集包含 9 个基准测试，其中八个要求以自由形式文本作答，

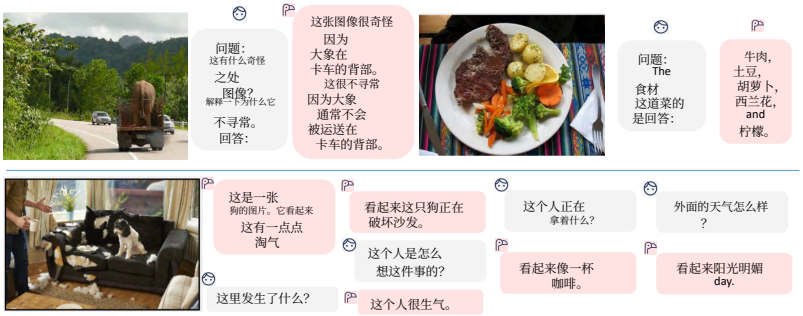


图 7.24 Flamingo 可以理解图像并用文本描述图像。灰色框表示用户输入，粉色框表示 Flamingo 输出。上排中，Flamingo 回答有关图像的问题。下排中，则是围绕一张照片展开的对话。图像改编自 [3, 第 31]页和 [3,第 32],页，经作者许可转载。



图 7.25 Flamingo 回答有关视频的问题。图中展示了部分视频帧。灰色框表示用户输入，粉色框表示 Flamingo 输出。图像改编自 [3, p. 33], ，经作者慷慨许可转载。

另一个基准测试（Kinetics 700）则需要进行分类。在这八个自由形式基准测试上，Flamingo 均能提升少样本最先进水平，且往往优势巨大。在其中四个基准测试上，Flamingo 的表现甚至超过了微调后的结果。这一点尤为引人注目，因为 Flamingo 仅使用了 32 个特定任务示例，其特定任务训练数据比当前最先进水平所用的数据少约 1,000 倍。

Flamingo 可以针对特定基准测试进行微调，以提升性能。微调过程中，冻结的模型部分不会改变。在 9 个示例任务上进行微调后，Flamingo 在其中 5 个任务上的微调后最先进水平得以提升。这表明，通过微调模型的 10B 自由参数，许多情况下都能将性能提升至新的水平。

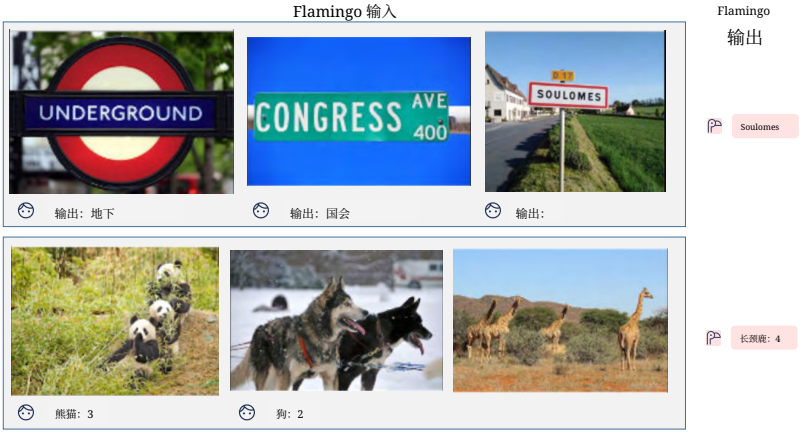


图 7.26 通过结合图像与文本对 Flamingo 进行少样本查询。请注意，在第二个示例中，Flamingo 没有统计树木，而是继续统计动物。少样本查询通常为 32 个。图像改编自 [3, 第 2], 页，经作者许可转载。

7.3.4 从文本生成视频

根据文本描述创作视频是一项重要课题，例如可用于教育或动态内容的呈现。虽然已有不少模型能够通过文本描述图像和视频，但关于反向生成的方案却不多。文本和视频的编码概念与视频图像描述相似。生成视频的质量可以通过多种指标来评估，这些指标会比较真实视频与生成视频之间的相似度。*FVD*（弗雷歇视频距离）是 Fréchet Inception 距离（FID）（第 7.2.6 节）的时空扩展，对视觉质量、时间连贯性和样本多样性十分敏感。

视频 Transformer [172] 将一维 Transformer 编码器-解码器扩展到视频领域。视频表示为 $\boldsymbol{x} \in \mathbb{R}^{h \times w \times s \times d}$，其中 h 和 w分别表示空间高度和宽度上的词元数量，s 表示时间轴上的词元数量，而 d 表示通道数（例如颜色）。视频被划分为时间和空间上的小型三维块。每个块分别应用自注意力。为了实现块之间的信息直接交换，各层的块大小有所变化。每个块包含 4 帧，空间分辨率为 32×32。不同层和维度中的自注意力范围为 1 到 32。最大模型的隐藏层维度为 2,048，包含 8 层和 373M 个参数。在 BAIR RobotPushing 数据集 [44] 上，该模型取得了 *FVD*（弗雷歇视频距离）得分 [167] 94，是发表时的最先进水平。

NÜWA [175] 是一种近期提出的 Transformer 编码器-解码器模型，可用于解决从文本生成视频的问题。它采用一种名为*3D*邻近注意力的机制，以捕捉空间轴和时间轴上的局部性特征。图像、视频和文本数据均表示为词元 $\boldsymbol{x} \in \mathbb{R}^{h \times w \times s \times d}$，其中$h$ 和 w



图 7.27 Imagen video [66]根据文本提示（下方）生成的视频。该模型能够生成多样且时间连贯的视频，并且与给定请求高度匹配。经作者 [66, p. 2]友好许可转载图片。



图 7.28 256×256 NÜWA [175] 根据图像上方的文本生成的图像，用于 MS COCO 基准测试。经作者 [175, p. 5]友好许可转载图片。

表示空间高度和宽度上的词元数量，s 表示时间轴上的词元数量，d 表示每个词元的维度。原始输入区域通过可训练的 VQ-GAN（第 7.2.3 节）转换为图像块的离散词元。与 CogView（第 7.2.6 节）使用的 VQ-VAE 相比，这个基于 GAN 的量化模块能提供更好的图像质量。

该模型修改了注意力计算，并针对宽度、高度和时间范围考虑局部邻域，这种机制称为 3D 邻近自注意力。宽度、高度和时间分别使用三种不同的位置编码嵌入。每个 336×336 像素视频帧被划分为 21×21 个图像块，并以每秒 2.5 帧的速度从视频中采样 10 帧。宽度、高度和时间上的邻域大小均为 3。该模型在三个任务上进行预训练：使用来自 Conceptual Captions 的 290 万个文本-图像对进行文本到图像生成；使用 Moments in Time 的 72.7 万个视频进行视频预测；以及使用 24.1 万个文本-视频对进行文本到视频生成。

对于文本到图像生成，NÜWA 在 MS COCO 数据集上进行微调。每段文本生成 60 张图像，并由 CLIP 选出最佳图像（第 7.2.4 节）。如图 7.28 所示，NÜWA 的 FID-0 为 12.9，优于 CogView，表现不错，但不及 LAFITE（8.1）和 OFA（10.5）。对于文本到视频，NÜWA

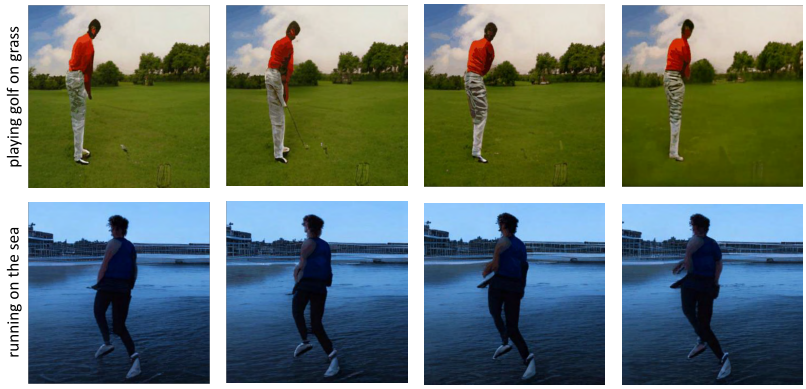


图 7.29 NÜWA 根据文本（左）生成的两个视频帧 [175]，用于 Kinetics 数据集上的文本到视频任务。请注意，模型从未见过诸如“在海上奔跑”这样的输入文本。经作者 [175, 许可转载， 5]页。

在 Kinetics 数据集上进行微调。图 7.29 展示了两个生成示例中的一些帧。NÜWA 在 FID-img 和 FID-vid 指标上表现最佳，得分分别为 28.5 和 7.0。视频预测需要根据视频的起始帧生成后续帧序列。在 BAIR 机器人推送数据集上，NÜWA 在这项任务中以 86.9 的 FVD 分数创下新的最先进水平。

NÜWA 还支持许多其他任务。在图像编辑方面，它可以重建图像的部分内容；也可以根据文本编辑标记的图像区域，例如“一匹马正在草地上奔跑”。带有文本标注的图像草图可以转换为照片。这种模式也可以应用于视频，即根据一系列带有标注区域的图像生成视频。最后，它还可以更改视频中的内容，例如修改图 7.30 下排所示潜水员的动作。此外，一系列带有文本标注的图像草图也可以转换为视频。更多示例如此处所示 [174]。

GODIVA [176]是同一批作者提出的类似先前方法，基于 VQ-VAE 变分自编码器。

Imagen Video 是近期推出的一种基于 Imagen 的高清文本到视频模型（图 7.17）。通过 Frozen T5 文本编码器-解码器和知识库视频扩散模型生成低分辨率视频。随后，级联视频扩散模型交替提升空间和时间分辨率 [66]，以 24 帧每秒的速率构建 128 个逼真的视频帧，分辨率为 1280×768。图 7.27 展示了 Imagen Video 根据文本提示生成的视频。

可用实现

- <style 印度尼西亚语='2'>VideoBERT 代码 https://github.com/ammesatyajit/VideoBERT
- COOT 代码 https://github.com/gingsi/coot-videotext

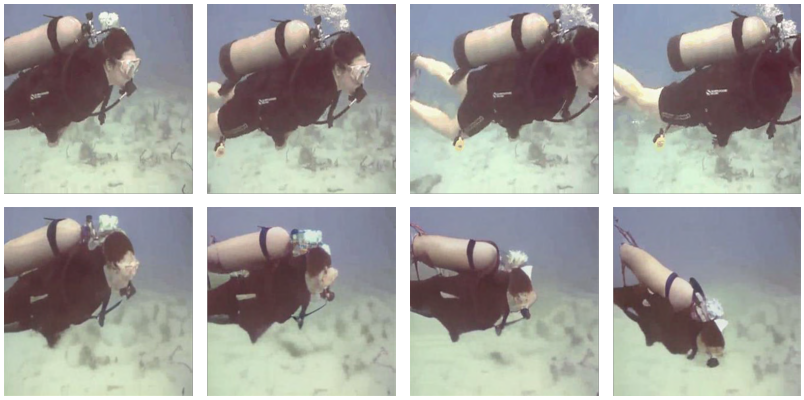


图 7.30 NÜWA [175] 可以编辑视频。上排显示原始视频。下排中，NÜWA 接收输入“潜水员正游向底部”，并据此修改视频。图片经作者许可转载 [175, 第 28] 页。

- DeCEMBERT 代码 https://github.com/zinengtang/decembert
- VATT 代码https://github.com/google-research/google-research/tree/master/vatt
- Omnivore 代码https://github.com/facebookresearch/omnivore
- 视频 Transformer 代码https://github.com/rakhimovv/lvt
- MTV 代码和模型https://github.com/google-research/scenic
- NÜWA 代码 https://github.com/lucidrains/nuwa-pytorch

7.3.5 总结

视频处理需要整合不同的模态，例如图像、以视频字幕形式呈现的文本，以及可能由自动语音识别转换为文本的语音。视频处理为图像处理增加了一个额外的时间维度。此外，深度信息和摄像机运动也可能很重要。自2019年以来，使用自监督预训练的大规模Transformer一直是视频处理领域的主流模型。这些模型可以解决不同的任务，例如视频描述生成、动作识别、视频问答、根据文本生成视频、下一帧预测、视频检索、音视语音识别等。

现有的跨模态预训练基础模型主要聚焦于：（1）改进模型架构；（2）利用更多数据；（3）设计更好的预训练任务。由于预训练语言模型的输入长度有限，视频必须划分为适当的词元。这种处理方式从对 30 多个片段进行聚合（VideoBERT）、使用固定视频块（VATT），到使用维度可变的视频块（COOT、MTV、Video Transformer）不等。一些模型（VideoBERT、DeCEMBERT）使用 CNN卷积生成低级特征。更常见的做法是使用 VQ-VAE 自编码器或基于生成对抗网络的 VQ-GAN 进行聚合。有时，视频和文本会分别使用预训练语言模型

处理，之后再合并（VATT）。另一种做法是将视频词元和文本词元拼接起来，再由单个预训练语言模型处理（Omnivore、Merlot）。Transformer 使用注意力机制处理空间维度和时间维度，通常会对其进行局部化，以减少计算开销。

不同模态的融合至关重要。文本与语言通过预训练任务建立关联：利用另一种模态的词元，预测掩码视频或文本词元。CoVeR 表明，当模型同时针对视频任务和图像任务进行微调时，性能可以得到提升。甚至还可以结合音频、文本和视频词元。

视频分析模型的性能取得了显著进展。在一年内，使用基础模型后，Kinetics-400 基准测试上的动作分类错误率降至 10.9%，下降了 33%。尽管取得了显著进步，最先进的方法仍无法提取或捕捉视频中包含的所有复杂时空信息。要理解视频中视觉内容的多样性以及相关文本描述的结构，仍有大量工作要做。

根据图像描述生成视频尚处于早期阶段，目前只能生成非常短的高清视频。然而，与 GPT-3 或 Gopher 等基础模型相比，当前模型规模相对较小。因此，可以预期，参数更多的模型将在性能上取得显著提升。

通用模型正成为一种趋势，例如 Nüwa 能够处理多种模态的数据并解决多种任务。利用不同媒体进行训练，可以相互促进模型在不同任务上的性能。拥有 80B 个参数的 Flamingo 基于一个大型预训练语言模型和一个单独预训练的视觉编码器。它能够处理由图像、文本和视频组成的混合序列。通过构建适配器模块和交叉注意力层，语言模型可以纳入视觉模态的结果，并执行多种分析任务，例如视觉问答、图像描述等。此外，无需专门微调，只需使用少样本提示，就能指示它解决许多任务。

尽管 Flamingo 无法生成与图像描述相对应的图像或视频，但它朝着多模态基础模型迈进了一步；这类模型有望成为多媒体处理的通用工具。通过少样本提示，它们能够解决数千甚至数百万项任务。由于可以结合针对不同媒体独立开发的理念，预计这一领域将取得重大进展。未来的发展方向还包括使用规模更大的训练数据，不过现有数据规模已经相当庞大。此外，根据该领域当前的研究现状，开发多语言视频模型也是顺理成章的结果。

7.4 动态系统控制

基础模型可以处理多种类型的序列。其中包括序贯决策问题：施事必须根据状态选择动作，并

获得奖励，同时生成下一个状态。这个过程必须重复多次，直到最终奖励总和确定。随后，任务是根据状态选择动作，使回报总和最大化。这个目标可以表述为一个序列问题，并可使用预训练语言模型来预测下一个最优动作。

7.4.1 决策变换器

预训练语言模型能够预测序列，例如文本词元或视频帧。遵循这一模式，预训练语言模型也能够对任意状态的演变进行建模。强化学习 考虑一个系统，其中包含状态 s_t、动作 a_t和奖励 $r_t = R(s_t, a_t)$，它们均处于给定的时间步 t。状态和奖励由环境决定，而动作必须由施事指定。强化学习的目标是学习一种策略 $a = \pi(s_t)$，使预期回报总和 $E(\sum_{t=1}^{T} r_t)$ 最大化。在线强化学习期间，可以访问环境；给定 (s_t, r_t, a_t) 后，环境会返回下一个状态 (s_{t+1}, r_{t+1})。离线强化学习中，只有一组有限的环境观测轨迹。后一种情形更难，因为施事无法再探索环境。

决策 Transformer [23] 运行于离线强化学习设置中。决策 Transformer 不直接使用回报 r_t，而是考虑前向奖励总和 $\hat{R}_t = \sum_{t'=t}^{T} r_{t'}$。因此，一条轨迹表示如下

$$\tau = \left(\hat{R}_1, s_1, a_1, \hat{R}_2, s_2, a_2, \ldots, \hat{R}_T, s_T, a_T\right) \tag{7.5}$$

$(s_t、 r_t、 a_t)$ 的输入词元嵌入通过线性层计算，每种模态使用不同的线性层（图 7.31）。如果状态是图像，则使用卷积编码器而不是线性层对其进行变换。随后，通过层归一化对这些嵌入进行归一化。对于包含三个输入的每个时间步，都会学习一个位置嵌入，并将其添加到该时间步的嵌入中。然后，这些嵌入由自回归 GPT 模型进行处理，该模型通过自回归建模预测未来动作。

训练基于一个由观测轨迹组成的数据集。从这些轨迹中采样长度为 K 的小批次。然后，GPT 模型针对每个$t = 1、\ldots\ldots、 K$，给定截至 s_t 的轨迹，预测 a_t。对于离散动作，使用交叉熵损失函数，目标是在时间 t提高实际动作的概率。对于连续动作，例如速度，则使用均方误差作为损失，以最小化其与观测控制值之间的平方差。无需预测状态或前向奖励总和。

对于初始状态 s_1，会指定一个基于期望性能（甚至是可能的最大回报）的目标前向奖励总和 $\hat{R}_1$。生成的动作 a_1 执行后，目标回报会减去所获得的奖励，并确定下一个状态 s_2。生成动作并

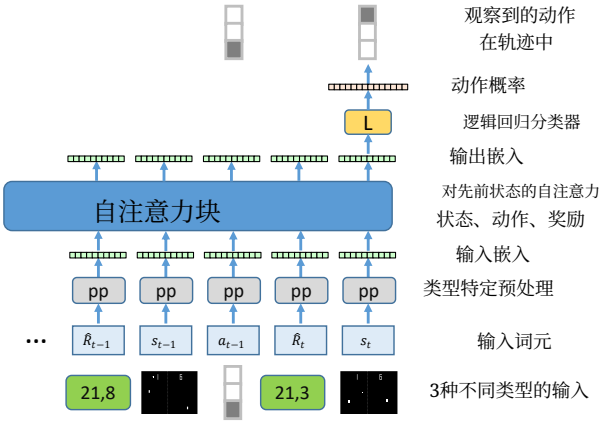


图 7.31 决策 Transformer 将自回归语言模型应用于奖励的前向累积和 $\hat{R}_t$、状态 s_t 和动作 a_t。状态也可以用视频帧的形式提供，例如乒乓球游戏中的视频帧。给定奖励的前向累积和 [23] 后，模型会预测轨迹中的下一个动作。

将其用于获取下一个前向奖励总和和下一个状态的过程会不断重复，直至轨迹结束。请注意，实际的前向奖励总和应接近期望性能的指定值。尽管模型只使用随机选取的子序列进行训练，但它可以学习“合并”来自不同训练轨迹的子序列，从而在测试时生成最优轨迹。显然，训练期间必须评估大量子序列，才能获得良好的解。

Atari 基准 [13] 包含离散动作，使用四个视频帧描述状态，并通过卷积编码器处理这些帧。这里只使用了 1% 的可用数据。在四项 Atari 任务（Breakout、Qbert、乒乓球和 Seaquest）中，通常会考虑长度为 $K = 30$ 的上下文。除 Qbert 外，决策 Transformer 的表现可与最先进的方法相媲美，并在两款游戏（Breakout、Seaquest）中取得最佳结果。最有效的替代方案是*CQL* [87]Q 学习器。

在*D4RL* 基准中，模拟了简单的机器人（HalfCheetah、Hopper 和 Walker），它们由连续值动作控制。在该基准测试中，决策 Transformer 在大多数情况下都比其他方法取得更好的结果，且平均性能最高。同样，CQL 是最佳替代方法。

作者评估了各类方法在一种环境中的性能，在这种环境中，奖励需要经过很长一段时间才能传播。*Key-to-Door*基准 [104] 分为三个阶段：

- 第一阶段，施事被放置在一个有钥匙的房间里；
- 随后，施事被放置在一个空房间里；
- 最后，施事被放置在一个有门的房间里。

施事在第三阶段到达门时会获得二元奖励，但前提是他在第一阶段拿到了钥匙。在这一基准测试中，决策

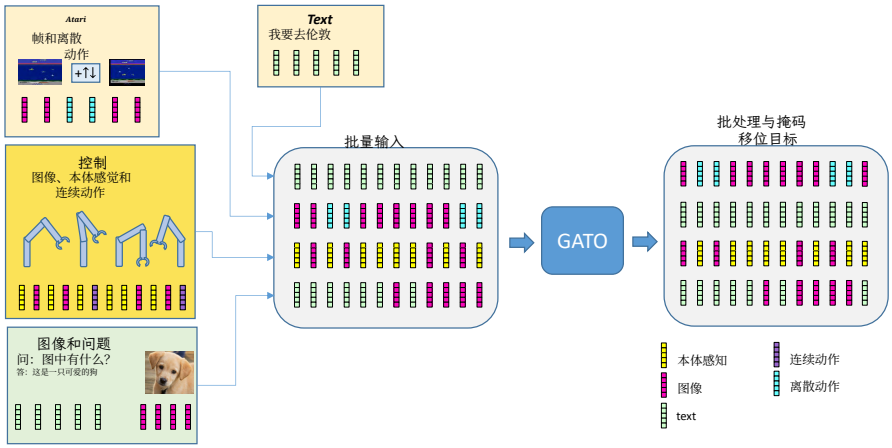


图 7.32 来自不同任务和模态的数据会被转换为序列，例如雅达利游戏中的帧和动作、文本词元序列、图像块词元，以及连续感觉输入与输出。在 Gato [134, 135], 中，一个大型仅解码器 Transformer 会处理该序列。训练期间，会使用特定变量（例如动作）来计算损失。图像改编自 [135, , 图片2], 鸣谢信息见表 A.3。

Transformer 及相关方法明显优于 Q 学习方法，后者无法在时域内有效传播奖励。

Reid 等人 [136] 对决策 Transformer 的详细设计进行了修改，从而提升了性能。Kumar 等人 [86] 通过理论分析表明，离线强化学习（决策 Transformer 采用的方法）在长时域任务上比简单地克隆专家行为具有更好的保证。这一点在数据噪声足够大的情况下尤其成立。

7.4.2 用于文本、图像和控制的 GATO模型

GATO [134] 是一个基础模型，使用约 600 种不同任务进行训练，涵盖文本生成、图像描述、用机械臂堆叠实体积木，以及玩雅达利主机游戏。根据上下文，它会自行决定输出哪些词元：文本、关节扭矩、按键输入，或其丰富输出能力中的其他形式。

输入会根据模态进行分词

- 文本通过 SentencePiece 编码，使用 32,000 个词元。
- 图像被转换为一系列不重叠的 16×16 图像块，类似于视觉Transformer（第 7.2.2 节）。

- 离散值（例如 Atari 按钮操作）按行优先顺序展平为整数序列。分词后的结果是一个取值范围在 $[0, 1024]$内的整数序列。

- 连续值，例如本体感知输入（对自主运动、力和身体位置的感知）或关节扭矩，会被预处理并离散化为 1,024 个区间。随后，这些离散整数会被移位到 32000、...、33024 的范围内。

属于文本、离散值或连续值观察结果或任意时间步动作的词元，使用查找表嵌入到经过学习的向量嵌入空间中。根据词元在相应时间步内的局部词元位置，为所有词元添加经过学习的位置编码。属于任意时间步图像块的词元则使用单个 ResNet [63]块进行嵌入，从而为每个图像块获得一个向量。此外，还会添加一个可学习的图像内位置编码向量（图 7.32）。

Gato 由一个拥有 12 亿参数、包含 24 层且嵌入维度为 2,048 的仅解码器 Transformer 组成。与每个语言模型一样，所有词元都会被预测，因此都可以设为训练目标。目前，仅将文本词元、离散值和连续值以及动作用作目标。通常，在训练期间必须最大化观测到的目标词元的概率。

为使 GATO 专注于特定任务，会使用来自同一源智能体在同一任务中生成的回合的提示。GATO 在 596 项不同的控制任务上接受了训练，其中包括 Atari 基准 [13]。作者只纳入了“良好”轨迹，其回报至少达到该任务专家回报的 80%。此外，GATO 还接受了 8 项视觉与语言任务的训练，例如使用 MS-COCO Captions [26] 和 Conceptual Captions [153] 进行图像描述，以及视觉问答数据集。此外，GATO 还使用包含 3000 亿个文本词元的大型 MassiveText[128] 进行训练。

GATO 在多种应用中的性能都经过了评估。在 Atari 基准测试中，该模型在 51 款雅达利游戏中的 23 款里达到了平均人工水平或更高的得分。在机器人堆叠基准测试中，GATO 的表现与 BC-IMP 基线 [90]相当。该模型的对话和图像描述功能仅处于初级水平，这并不令人意外，因为其模型规模较小。

GATO模型是首次尝试使用同一个基础模型同时解决文本、图像和控制任务。在控制任务中，它取得了尚可的结果，而在文本和图像任务中的性能则仅属一般。或许，决策 Transformer 的前向奖励总和表示能够使其受益。实际的基础模型拥有数千亿个参数，并需要相应的计算开销。如果将 GATO 模型扩展到这一规模，可以预期其性能也会相应提升。

可用实现

- 决策 Transformer 代码 https://sites.google.com/berkeley.edu/decision-transformer

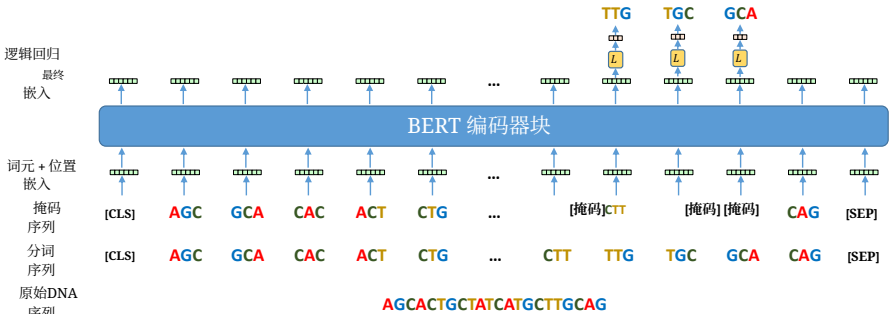


图 7.33 DNABERT将DNA序列分词为相互重叠的3元组，并训练标准BERT模型来预测掩码词元 [76]。由此得到的模型可微调以执行多种DNA解读任务。

7.4.3 总结

预训练语言模型可以应用于包含多种元素类型的序列。决策 Transformer 处理在序贯决策问题（例如电子游戏、机器人控制或自动驾驶）过程中，在特定时间步出现的奖励、状态和动作序列。它对这些量的观测轨迹进行建模。它不将奖励作为输入，而是考虑截至轨迹末尾的奖励总和，因为这是需要最大化的量。对于每种输入类型，都会执行一些预处理以生成嵌入。决策 Transformer 经过训练，可以预测由 30 个时间步组成的短子序列中的动作。

在应用过程中，可以将期望的未来回报总和设为一个条件。随后，模型能够拼接训练数据中不同子序列的信息，从而获得近乎最优的动作，使奖励总和达到最大值。大量针对不同基准测试的实验验证了这一点。

GATO模型表明，预训练语言模型也可以同时用于解决强化学习任务、文本任务和图像任务。该模型使用近600项控制基准测试、8项图像任务以及3000亿个文本词元进行训练。该模型的文本和图像描述能力还很基础，但在 Atari 基准上的性能相对不错。这只是一个概念验证；通过增大模型规模，以及例如采用前向奖励总和，可以进一步提升模型表现。

7.5 DNA与蛋白质序列的解读

破译DNA的语言是生物学研究最重要的目标之一。遗传密码具有普遍性，它解释了DNA如何被翻译成蛋白质。相比之下，决定基因何时以及如何表达的调控密码，会因不同的细胞类型和生物体而异。这类似于自然语言文本中的一词多义和远距离语义关系。**DNABERT**[76]将DNA序列切分为相互重叠的3元组，并训练标准BERT模型来预测掩码词元（图7.33）。在大量DNA序列上进行预训练后，通过针对许多特定DNA预测任务进行微调，它可以提升最先进水平。这些任务包括序列基序分析（具有生物学相关性的DNA片段）和启动子区域预测（能够使基因表达受到调控的核苷酸序列）。MoDNA [5] 和 GeneBERT [106] 具有类似的功能。

蛋白质是由共价键连接的氨基酸线性链。氨基酸可以用包含 25 个字符的字母表表示。这些字符串非常适合采用多种自然语言处理方法 [111]。**AminoBERT** [29] 是一种语言模型，以蛋白质序列为输入，预测三维蛋白质结构。它还采用一种自然的方法来描述多肽几何结构，这种方法在整个多肽层面上不受旋转和平移的影响。平均而言，该模型在孤儿蛋白质和工程化蛋白质类别上的表现优于 AlphaFold2[80] 和 RoseTTAFold [8]，计算时间最多可缩短 106 倍。

还有一些模型能取得类似的结果 [97]，例如蛋白质语言模型 **ESMFold**。它会生成可用于下游任务的嵌入，例如用于捕捉蛋白质的结构特性。一个拥有 15B 个参数的模型可以预测蛋白质的三维结构，精度达到单个原子级别。

可用的实现

- DNABERT代码和模型 https://github.com/jerryji1993/DNABERT
- GeneBERT代码和模型 https://github.com/ZovcIfzm/GeneBERT/tree/main/GeneBERT
- ProteinBERT代码和模型 https://github.com/nadavbra/protein_bert
- AlphaFold 2代码和模型 https://github.com/deepmind/alphafold
- RoseTTAFold代码和模型 https://github.com/RosettaCommons/RoseTTAFold
- ESMFold代码和模型 https://github.com/facebookresearch/esm

7.5.1 总结

基础模型还可以应用于 DNA 和蛋白质序列，以提取序列元素的上下文嵌入。通过这种方法，模型能够积累关于这些序列的大量知识，并达到最先进水平

的性能，在各种下游任务中大幅超越现有工具。这些模型可以帮助预测蛋白质的三维结构。这对于其功能至关重要，并可能有助于开发能够影响其功能的活性物质。

参考文献

[1] T. Afouras、J. S. Chung 和 A. Zisserman。《LRS3-TED：用于视觉语音识别的大规模数据集》。2018年。arXiv：1809.00496。[2] H. Akbari、L. Yuan、R. Qian、W.-H. Chuang、S.-F. Chang、Y. Cui 和 B. Gong。《VATT：基于原始视频、音频和文本进行多模态自监督学习的 Transformer》。2021年12月6日。arXiv：2104.11178 [cs，eess]。[3] J.-B. Alayrac 等人。Flamingo：用于少样本学习的视觉语言模型。4月 2022 10 48550 2204 14198 2204 14198 [ ] 29, 。doi：。/ arXiv。。arXiv：。cs。[4] P. Ammanabrolu 和 M. Riedl。《学习文本环境的基于知识图谱的世界模型》。载于：神经信息处理系统进展 34（2021年），第 3720–3731 页。[5] W. An、Y. Guo、Y. Bian、H. Ma、J. Yang、C. Li 和 J. Huang。《MoDNA：面向基序的 DNA 语言模型预训练》。载于：第 13 届 ACM 生物信息学、计算生物学与健康信息学国际会议论文集。BCB '22。美国纽约州纽约市：计算机协会，2022年8月7日，第 1–5 页。ISBN：978-1-4503-9386-7。doi：10.1145/3535508.3545512。[6] P. Anderson。VQA2VLN 2021教程。从 VQA 到 VLN：视觉与语言研究的最新进展。2021年6月20日。网址：https:// vqa2vln- tutorial.github.io/（访问于2022年3月25日）。[7] I. Anokhin、K. Demochkin、T. Khakhulin、G. Sterkin、V. Lempitsky 和 D. Korzhenkov。《采用条件独立像素合成的图像生成器》。载于：IEEE/CVF 计算机视觉与模式识别会议论文集。2021年，第 14278–14287 页。[8] M. Baek 等人。《利用三轨神经网络准确预测蛋白质结构与相互作用》。载于：《科学》373.6557（2021年8月20日），第 871–876 页。doi：10.1126/science.abj8754。[9] A. Baevski、W.-N. Hsu、Q. Xu、A. Babu、J. Gu 和 M. Auli。《Data2vec：用于语音、视觉和语言自监督学习的通用框架》。2022年1月22日。arXiv：2202.03555。[10] A. Baevski、H. Zhou、A. Mohamed 和 M. Auli。《Wav2vec 2.0：语音表征自监督学习框架》。2020年。arXiv：2006.11477。[11] H. Bao、L. Dong 和 F. Wei。《BEiT：图像 Transformer 的 BERT 预训练》。2021年。arXiv：2106.08254。[12] R. Beaumont。LAION-5B：开放式大规模多模态数据集的新时代 | LAION。2022年8月8日。网址：https://laion.ai/blog/laion-5b（访问于2022年8月29日）。[13] M. G. Bellemare、Y. Naddaf、J. Veness 和 M. Bowling。《街机学习环境：通用智能体的评估平台》。载于：《人工智能研究杂志》47（2013年），第 253–279 页。[14] Ş. Bilici。《深度神经网络音乐生成综述》。Safak的博客。2020年10月15日。网址：https : / / safakkbilici . github . io / a - survey - on - music - generation/（访问于 2022 年 3 月 3 日）。[15] A. Blattmann、R. Rombach、K. Oktay 和 B. Ommer。检索增强扩散模型。2022年4月26日。doi：10.48550/arXiv.2204.11824。arXiv：2204.11824 [cs]。[16] A. Brock、S. De、S. L. Smith 和 K. Simonyan。《无需归一化的高性能大规模图像识别》。2021年。arXiv：2102.06171。[17] S. Cable。《Alexa，用我祖母的声音给我读这本书》。载于：新闻（2022年6月24日 0140 0460）。ISSN：-。网址：https://www.thetimes.co.uk/article/alexa-read-me-this-book-in-my-grandmothers-voice-cfdtjbjcc（访问于2022年7月8日）。

[18] R. Cai、J. Yuan、B. Xu 和 Z. Hao。《SADGA：用于文本到SQL的结构感知双图聚合网络》。载于：神经信息处理系统进展 34（2021年），第 7664–7676 页。[19] J. Cao、Z. Gan、Y. Cheng、L. Yu、Y.-C. Chen 和 J. Liu。《幕后：揭示预训练视觉语言模型的秘密》。载于：欧洲计算机视觉会议。施普林格，2020年，第 565–580 页。[20] Y.-H. Cao、H. Yu 和 J. Wu。《仅用 2040 张图像训练视觉 Transformer》。2022年1月25日。arXiv：2201.10728 [cs]。[21] W. Chan、D. Park、C. Lee、Y. Zhang、Q. Le 和 M. Norouzi。《SpeechStew：混合所有可用语音识别数据来训练一个大型神经网络》。2021年。arXiv：2104.02133。[22] H. Chefer、S. Gur 和 L. Wolf。《超越注意力可视化的 Transformer 可解释性》。载于：IEEE/CVF 计算机视觉与模式识别会议论文集。2021年，第 782–791 页。[23] L. Chen 等人。《决策 Transformer：通过序列建模实现强化学习》。载于：神经信息处理系统进展 34（2021年）。[24] S.-J. Chen、A. S. Subramanian、H. Xu 和 S. Watanabe。《利用 CHiME-4挑战赛和语音增强基线配置构建最先进的远场语音识别系统》。2018年。arXiv：1803.10109。[25] W. Chen、M.-W. Chang、E. Schlinger、W. Wang 和 W. W. Cohen。《基于表格和文本的开放域问答》。2020年。arXiv：2010.10439。[26] X. Chen、H. Fang、T.-Y. Lin、R. Vedantam、S. Gupta、P. Dollár 和 C. L. Zitnick。《Microsoft COCO图像描述：数据收集与评测服务器》。2015年。arXiv：1504. 00325。[27] R. Child、S. Gray、A. Radford 和 I. Sutskever。《使用稀疏 Transformer 生成长序列》。2019年。arXiv：1904.10509。[28] J. Cho、J. Lu、D. Schwenk、H. Hajishirzi 和 A. Kembhavi。《X-LXMERT：利用多模态 Transformer 绘画、生成图像描述并回答问题》。2020年。arXiv：2009.11278。[29] R. Chowdhury、N. Bouatta 和 S. Biswas。《利用语言模型和深度学习进行单序列蛋白质结构预测》。载于：《自然·生物技术》（2022年10月3日），第 1–7 页。网址：https://www.nature.com/articles/s41587-022-01432-w（访问于2022年10月14日）。[30] Y.-S. Chuang、C.-L. Liu、H.-Y. Lee 和 L.-s. Lee。《SpeechBERT：端到端口语问答的音频与文本联合学习语言模型》。8月 11, 2020 1910 11559 [ ] 。arXiv：。cs, eess。[31] Y.-A. Chung、Y. Zhang、W. Han、C.-C. Chiu、J. Qin、R. Pang 和 Y. Wu。《W2v-BERT：结合对比学习与掩码语言建模进行自监督语音预训练》。2021年。arXiv：2108.06209。[32] coco。Papers with Code——COCO图像描述基准（图像描述）。2022年3月6日。网址：https://paperswithcode.com/sota/image-captioning-on-coco-captions（访问于2022年3月6日）。[33] D. Damen 等人。《重新审视自我中心视觉：EPIC-KITCHENS-100 的数据收集、流程与挑战》。载于：《国际计算机视觉杂志》130.1（2022年），第 33–55 页。[34] K. Desai、G. Kaul、Z. Aysola 和 J. Johnson。《RedCaps：由大众为大众创建的网络精选图文数据》。2021年。arXiv：2111.11431。[35] P. Dhariwal。OpenAI Jukebox样本浏览器。2020年。网址：https://jukebox.openai.com/（访问于2022年3月3日）。[36] P. Dhariwal、H. Jun、C. Payne、J. W. Kim、A. Radford 和 I. Sutskever。《Jukebox：一种音乐生成模型》。2020年4月30日。arXiv：2005.00341 [cs，eess，stat]。[37] P. Dhariwal 和 A. Nichol。《扩散模型在图像合成方面胜过生成对抗网络》。载于：神经信息处理系统进展 34（2021年）。[38] S. Di 等人。《利用可控音乐 Transformer 生成视频背景音乐》。载于：第 29 届 ACM 多媒体国际会议论文集。2021年，第 2037–2045 页。[39] D. Ding、F. Hill、A. Santoro、M. Reynolds 和 M. Botvinick。《对学习得到的对象嵌入进行注意力处理，可实现复杂的视觉推理》。载于：神经信息处理系统进展 34（2021年）。[40] M. Ding 等人。《CogView：通过 Transformer 掌握文本到图像生成》。11月 5, 2021 2105 13290 [ ] 。arXiv：。cs。

[41] A. Dosovitskiy 和 T. Brox。《基于深度网络的感知相似度指标生成图像》。载于：神经信息处理系统进展 29（2016年）。[42] A. Dosovitskiy 等。《一张图像胜过 16×16 个词：用于大规模图像识别的 Transformer》。2020年。arXiv：2010.11929。[43] Y. Du、Z. Liu、J. Li 和 W. X. Zhao。《视觉语言预训练模型综述》。2022年。arXiv：2202.10936。[44] F. Ebert、C. Finn、A. X. Lee 和 S. Levine。《借助时间跳跃连接进行自监督视觉规划》。载于：CoRL。2017年，第344–356页。[45] P. Esser、R. Rombach 和 B. Ommer。《驾驭 Transformer，实现高分辨率图像合成》。载于：IEEE/CVF 计算机视觉与模式识别会议论文集。2021年，第12873–12883页。[46] N. Fei 等。《WenLan 2.0：借助多模态基础模型让人工智能拥有想象力》。2021年。arXiv：2110.14378。[47] W. Feller。《随机过程理论，特别论及其应用》。载于：第一届伯克利数学统计与概率研讨会论文集。加州大学出版社，1949年，第403–432页。[48] O. Gafni、A. Polyak、O. Ashual、S. Sheynin、D. Parikh 和 Y. Taigman。为人工智能图像生成提供更强的创作控制。2022 年 7 月 14 日。网址：https://ai.facebook.com/blog/greater-creative-control-for-ai-image-generation/（访问日期：2022年7月29日）。[49] L. Gao 等。《The Pile：用于语言建模的多样化文本数据集（800GB）》。2020年。arXiv：2101.00027。[50] K. Gavrilyuk、R. Sanford、M. Javan 和 C. G. Snoek。《用于群体活动识别的 Actor-Transformer》。载于：IEEE/CVF 计算机视觉与模式识别会议论文集。2020年，第839–848页。[51] S. Ging、M. Zolfaghari、H. Pirsiavash 和 T. Brox。《COOT：用于视频文本表征学习的协作式层次 Transformer》。2020年11月1日。arXiv：2011.00597。[52] R. Girdhar、M. Singh、N. Ravi、L. van der Maaten、A. Joulin 和 I. Misra。《Omnivore：适用于多种视觉模态的单一模型》。2022年。arXiv：2201.08377。[53] I. Goodfellow 等。《生成对抗网络》。载于：神经信息处理系统进展。2014年，第2672–2680页。[54] google。AVA：原子视觉动作视频数据集。2020年。网址：https://research.google.com/ava/（访问日期：2022年3月12日）。[55] R. Goyal 等。《用于学习和评估视觉常识的“Something Something”视频数据库》。载于：IEEE 国际计算机视觉会议论文集。2017年，第5842–5850页。[56] Y. Goyal、T. Khot、D. Summers-Stay、D. Batra 和 D. Parikh。《让视觉问答中的“视觉”发挥作用：提升图像理解在视觉问答中的作用》。载于：IEEE 计算机视觉与模式识别会议论文集。2017年，第6904–6913页。[57] A. Graves、S. Fernández、F. Gomez 和 J. Schmidhuber。《连接主义时序分类：利用循环神经网络为未分段序列数据标注》。载于：第23届国际机器学习会议论文集。2006年，第369–376页。[58] A. Gu、K. Goel 和 C. Ré。《利用结构化状态空间高效建模长序列》。2021年。arXiv：2111.00396。[59] A. Gulati 等。《Conformer：用于语音识别的卷积增强 Transformer》。2020年。arXiv：2005.08100。[60] Y. Guo 等。《从一般到具体：通过平衡调整生成信息丰富的场景图》。载于：IEEE/CVF 国际计算机视觉会议论文集。2021年，第16383–16392页。[61] K. Gupta、J. Lazarow、A. Achille、L. S. Davis、V. Mahadevan 和 A. Shrivastava。《Layout-transformer：利用自注意力进行布局生成与补全》。载于：IEEE/CVF 国际计算机视觉会议论文集。2021年，第1004–1014页。[62] A. M. Hafiz、S. A. Parah 和 R. U. A. Bhat。《机器视觉中的注意力机制与深度学习：最先进水平综述》。2021年6月3日。arXiv：2106.07550。[63] K. He、X. Zhang、S. Ren 和 J. Sun。《用于图像识别的深度残差学习》。载于：IEEE 计算机视觉与模式识别会议论文集。2016年，第770–778页。

[64] M. Heusel、H. Ramsauer、T. Unterthiner、B. Nessler 和 S. Hochreiter。《通过双时间尺度更新规则训练的生成对抗网络收敛于局部纳什均衡》。载于：神经信息处理系统进展 30（2017年）。[65] J. Ho、A. Jain 和 P. Abbeel。《去噪扩散概率模型》。载于：神经信息处理系统进展 33（2020年），第6840–6851页。[66] J. Ho 等。《Imagen Video：利用扩散模型生成高清视频》。2022年。arXiv：2210.02303。[67] M. Z. Hossain、F. Sohel、M. F. Shiratuddin 和 H. Laga。《深度学习在图像描述生成中的综合综述》。载于：ACM 计算调查 CsUR 51.6（2019年），第1–36页。[68] W.-N. Hsu、B. Bolte、Y.-H. H. Tsai、K. Lakhotia、R. Salakhutdinov 和 A. Mohamed。《Hubert：通过隐藏单元掩码预测进行自监督语音表示学习》。载于：IEEE/ACM 音频、语音与语言处理汇刊 29（2021年），第3451–3460页。[69] X. Hu、X. Yin、K. Lin、L. Wang、L. Zhang、J. Gao 和 Z. Liu。《Vivo：通过视觉词汇预训练，以新物体描述生成超越人类表现》。2020年。arXiv：2009.13682。[70] X. Hu、X. Yin、K. Lin、L. Zhang、J. Gao、L. Wang 和 Z. Liu。《VIVO：用于新物体描述生成的视觉词汇预训练》。载于：AAAI 人工智能会议论文集。第35卷第2期。2021年，第1575–1583页。[71] C.-Z. A. Huang 等。《Music Transformer：生成具有长期结构的音乐》。载于：国际学习表征会议（ICLR）。2019年。[72] Y. Huang、H. Xue、B. Liu 和 Y. Lu。《用于双向图像和文本生成的统一多模态 Transformer》。载于：第29届 ACM 国际多媒体会议论文集。2021年，第1138–1147页。[73] S. Islam、A. Dash、A. Seum、A. H. Raj、T. Hossain 和 F. M. Shah。《视频描述生成技术探索：深度学习方法综合综述》。载于：SN 计算机科学 2.2（2021年），第1–28页。[74] K. Ito 和 L. Johnson。LJ语音数据集。2017年。网址：https://keithito.com/LJ-Speech-Dataset（访问日期：2022年3月24日）。[75] E. Jang、S. Gu 和 B. Poole。《利用 Gumbel-Softmax 进行类别重参数化》。2016年。arXiv：1611.01144。[76] Y. Ji、Z. Zhou、H. Liu 和 R. V. Davuluri。《DNABERT：用于基因组中 DNA语言的 Transformer 模型预训练双向编码器表示》。载于：生物信息学 37.15（2021年），第2112–2120页。[77] C. Jia 和 Y. Yang。ALIGN：通过噪声文本监督扩展视觉和视觉语言表示学习。Google AI 博客。2021年5月11日。网址：http://ai.googleblog.com/2021/05/align-scaling-up-visual-and-vision.html（访问日期：2021年6月8日）。[78] Y. Jia。高质量、稳健且负责任的直接语音到语音翻译。Google AI 博客。2021年9月23日。网址：http://ai.googleblog.com/2021/09/high-quality-robust-and-responsible.html（访问日期：2021年10月25日）。[79] D. Jin、Z. Jin 和 R. Mihalcea。《文本属性迁移的深度学习：综述》。2020年。arXiv：2011.00416。[80] J. Jumper 等。《利用 AlphaFold 高精度预测蛋白质结构》。载于：《自然》596.7873（7873，2021年8月），第583–589页。ISSN：1476-4687。DOI：10.1038/s41586-021-03819-2。[81] T. Kano、S. Sakti 和 S. Nakamura。《基于 Transformer 和转码器的直接语音到语音翻译》。载于：（2021年）。[82] T. Karras、S. Laine、M. Aittala、J. Hellsten、J. Lehtinen 和 T. Aila。《分析并改进 StyleGAN 的图像质量》。载于：IEEE/CVF 计算机视觉与模式识别会议论文集。2020年，第8110–8119页。[83] W. Kay 等。《Kinetics人类动作视频数据集》。2017年。arXiv：1705.06950。[84] S. Khan、M. Naseer、M. Hayat、S. W. Zamir、F. S. Khan 和 M. Shah。《视觉中的 Transformer：综述》。载于：ACM 计算调查（2022年1月6日 0300, 1557 3505244 0360 7341 10 1145 3505244），第 页。ISSN：--。DOI：./。

[85] K. Khurana 和 U. Deshpande。《视频问答技术、基准数据集与评估指标：利用视频描述生成的全面综述》。载于：IEEE Access（2021年）。[86] A. Kumar、J. Hong、A. Singh 和 S. Levine。我们应在何时优先选择离线强化学习而非行为克隆? 2022年4月12日。arXiv：2204.05618 [cs]。[87] A. Kumar、A. Zhou、G. Tucker 和 S. Levine。《用于离线强化学习的保守Q学习》。载于：神经信息处理系统进展 33（2020年），第1179–1191页。[88] M. Kumar、D. Weissenborn 和 N. Kalchbrenner。《着色Transformer》。2021年。arXiv：2102.04432。[89] K. Lakhotia 等。《从原始音频进行生成式口语语言建模》。9月 9, 2021 2102 01192 [ ] 。

arXiv：. cs .[90] A. X. Lee 等。《超越抓取与放置：应对多样形状的机器人堆叠》。载于：第5届机器人学习年度会议。2021年。[91] H. Lee、U. Ullah、J.-S. Lee、B. Jeong 和 H.-C. Choi。《文本驱动图像生成与操控简述》。载于：2021年IEEE消费电子-亚洲国际会议（ICCE-Asia）。IEEE，2021年，第1–4页。[92] Z. Leng、M. Tan、C. Liu、E. D. Cubuk、J. Shi、S. Cheng 和 D. Anguelov。《PolyLoss：分类损失函数的多项式展开视角》。载于：国际学习表征会议。2021年。[93] M. Li 等。《CLIP-Event：通过事件结构连接文本与图像》。2022年。arXiv：2201.05078。[94] N. Li、S. Liu、Y. Liu、S. Zhao 和 M. Liu。《基于Transformer网络的神经语音合成》。载于：AAAI人工智能会议论文集，第33卷，第01期。2019年，第6706–6713页。[95] X. Li 等。《Oscar：面向视觉语言任务的对象语义对齐预训练》。载于：欧洲计算机视觉会议。施普林格，2020年，第121–137页。[96] J. Liang、J. Cao、G. Sun、K. Zhang、L. Van Gool 和 R. Timofte。《Swinir：使用Swin Transformer进行图像复原》。载于：IEEE/CVF计算机视觉国际会议论文集。2021年，第1833–1844页。[97] Z. Lin 等。《进化尺度上的蛋白质序列语言模型可实现准确的结构预测》。载于：bioRxiv（2022年）。[98] A. T. Liu、S.-W. Li 和 H.-y. Lee。《TERA：面向语音的Transformer编码器表征自监督学习》。载于：IEEE/ACM音频、语音与语言处理汇刊 29（2021年），第2351–2366页。[99] Z. Liu 等。《Swin Transformer：使用移位窗口的层次化视觉Transformer》。载于：IEEE/CVF计算机视觉国际会议论文集。2021年，第10012–10022页。[100] J. Lu、D. Batra、D. Parikh 和 S. Lee。《VilBERT：面向视觉与语言任务的任务无关视觉语言表征预训练》。载于：神经信息处理系统进展。2019年，第13–23页。[101] M. Malik、M. K. Malik、K. Mehmood 和 I. Makhdoom。《自动语音识别：综述》。载于：多媒体工具与应用（2020年），第1–47页。[102] M. Malik、M. K. Malik、K. Mehmood 和 I. Makhdoom。《自动语音识别：综述》。载于：多媒体工具与应用 80.6（2021年），第9411–9457页。[103] C. Mao、L. Jiang、M. Dehghani、C. Vondrick、R. Sukthankar 和 I. Essa。《离散表征增强视觉Transformer的鲁棒性》。2021年11月19日。arXiv：2111.10493 [cs]。[104] T. Mesnard 等。《无模型强化学习中的反事实信用分配》。2020年。arXiv：2011.09464。[105] A. Miech、D. Zhukov、J.-B. Alayrac、M. Tapaswi、I. Laptev 和 J. Sivic。《HowTo100M：通过观看一亿个带旁白的视频片段学习文本-视频嵌入》。2019年7月31日。arXiv：1906.03327 [cs]。[106] S. Mo 等。《面向不同细胞类型调控基因组的多模态自监督预训练》。2021年。arXiv：2110.05231。[107] M. Monfort 等。《Moments in Time数据集：用于事件理解的一百万个视频》。载于：IEEE模式分析与机器智能汇刊 42.2（2019年），第502–508页。

[108] M. M. Naseer、K. Ranasinghe、S. H. Khan、M. Hayat、F. Shahbaz Khan 和 M.-H. Yang。《视觉Transformer的有趣特性》。载于：神经信息处理系统进展 34（2021年）。[109] A. Nichol 等。《GLIDE：利用文本引导扩散模型实现照片级图像生成与编辑》。2021年。arXiv：2112.10741。[110] A. Q. Nichol 和 P. Dhariwal。《改进的去噪扩散概率模型》。载于：国际机器学习会议。PMLR，2021年，第8162–8171页。[111] D. Ofer、N. Brandes 和 M. Linial。《蛋白质语言：自然语言处理、机器学习与蛋白质序列》。载于：计算结构与生物技术期刊 19（2021年），第1750–1758页。[112] A. Oluwasammi 等。《从特征到文本：语义分割与图像描述深度学习的全面综述》。载于：复杂度 2021（2021年）。[113] A. van den Oord、O. Vinyals 和 K. Kavukcuoglu。《神经离散表示学习》。2018年5月30日。arXiv：1711.00937 [cs]。[114] A. van den Oord 等。《WaveNet：用于原始音频的生成模型》。2016年。arXiv：1609.03499。[115] OpenAI。DALL·E现已开启测试版。2022年7月20日。网址：https://openai.com/blog/dall-e-now-available-in-beta/（访问于2022年7月29日）。[116] V. Panayotov、G. Chen、D. Povey 和 S. Khudanpur。《LibriSpeech：基于公有领域有声读物的自动语音识别语料库》。载于：2015年IEEE声学、语音与信号处理国际会议（ICASSP）。IEEE，2015年，第5206–5210页。[117] I. Papastratis。语音识别：不同深度学习方法综述。AI Summer。2021年7月14日。网址：https://theaisummer.com/speech-recognition/（访问于2022年3月2日）。[118] papers-with-code。Papers with Code——ImageNet基准（图像分类）。2022年。网址：https://paperswithcode.com/sota/image-classification-on-imagenet（访问于2022年3月5日）。[119] K. K. Parida、S. Srivastava 和 G. Sharma。《从单声道到双耳：结合深度与跨模态注意力，从单声道音频生成双耳音频》。载于：IEEE/CVF冬季应用计算机视觉会议论文集。2022年，第3347–3356页。[120] D. S. Park、W. Chan、Y. Zhang、C.-C. Chiu、B. Zoph、E. D. Cubuk 和 Q. V. Le。《SpecAugment：一种用于自动语音识别的简单数据增强方法》。2019年。arXiv：1904.08779。[121] D. S. Park 等。《改进的自动语音识别噪声学生训练》。2020年。arXiv：2005.09629。[122] T. Park、M.-Y. Liu、T.-C. Wang 和 J.-Y. Zhu。《利用空间自适应归一化进行语义图像合成》。2019年11月5日。arXiv：1903.07291 [cs]。[123] C. Payne。《MuseNet》。载于：OpenAI博客（2019年）。[124] J. Perez-Martin、B. Bustos、S. J. F. Guimarães、I. Sipiran、J. Pérez 和 G. C. Said。《从视频转文本视角弥合视觉与语言：全面综述》。2021年。arXiv：2103.14785。[125] R. Prenger、R. Valle 和 B. Catanzaro。《WaveGlow：用于语音合成的基于流的生成网络》。载于：ICASSP 2019——2019年IEEE声学、语音与信号处理国际会议（ICASSP）。IEEE，2019年，第3617–3621页。[126] A. Radford、I. Sutskever、J. W. Kim、G. Krueger 和 S. Agarwal。CLIP：连接文本与图像。2021年1月5日。网址：https://openai.com/blog/clip/。[127] A. Radford 等。《从自然语言监督中学习可迁移视觉模型》。载于：国际机器学习会议。PMLR，2021年，第8748–8763页。[128] J. W. Rae 等。《扩展语言模型：Gopher训练方法、分析与见解》。载于：ArXiv预印本 ArXiv211211446（2021年12月8日），第118页。[129] c. raffel。*C4* | TensorFlow数据集。TensorFlow。2019年。网址：https://www.tensorflow.org/datasets/catalog/c4（访问于2021年12月14日）。[130] M. Raghu、T. Unterthiner、S. Kornblith、C. Zhang 和 A. Dosovitskiy。《视觉Transformer是否像卷积神经网络一样观察世界？》。载于：（2021年12月1日），第13页。[131] P. Ramachandran、B. Zoph 和 Q. V. Le。《探索激活函数》。2017年。arXiv：1710.05941。

[132] A. Ramesh、P. Dhariwal、A. Nichol、C. Chu 和 M. Chen。《基于CLIP潜变量的分层文本条件图像生成》。2022年4月12日。arXiv：2204.06125 [cs]。[133] A. Ramesh 等人。《零样本文本到图像生成》。2021年2月26日。arXiv：2102.12092。[134] S. Reed。《通用智能体》。2022年5月12日。网址：https://www.deepmind.com/publications/a- generalist-agent（访问于2022年5月19日）。[135] S. Reed 等人。《通用智能体》。2022年5月12日。arXiv：2205.06175 [cs]。[136] M. Reid、Y. Yamada 和 S. S. Gu。《维基百科能否助力离线强化学习？》2022年1月28日。arXiv：2201.12122 [cs]。[137] S. Ren、K. He、R. Girshick 和 J. Sun。《Faster R-CNN：借助区域提议网络实现实时目标检测》。2016年1月6日。arXiv：1506.01497 [cs]。[138] Y. Ren、C. Hu、X. Tan、T. Qin、S. Zhao、Z. Zhao 和 T.-Y. Liu。《FastSpeech 2：快速、高质量的端到端文本到语音》。2021年3月4日。arXiv：2006.04558。[139] M. Rivière 和 E. Dupoux。《迈向野外语音特征的无监督学习》。载于：2021年IEEE 口语语言技术研讨会（SLT）。IEEE，2021年，第156–163页。[140] J. Rodriguez。《悟道2.0的五个关键事实：有史以来构建的最大 Transformer 模型》。DataSeries。2021年9月21日。网址：https://medium.com/dataseries/five-key-facts-wu-dao-2-0-the-largest-transformer-model-ever-built-19316159796b（访问于2021年12月12日）。[141] R. Rombach。《潜扩散模型》。慕尼黑路德维希-马克西米利安大学 CompVis 机器视觉与学习实验室，2022年8月29日。网址：https://github.com/CompVis/latent-diffusion（访问于2022年8月29日）。[142] R. Rombach、A. Blattmann、D. Lorenz、P. Esser 和 B. Ommer。《基于潜在扩散的高分辨率图像合成……》。CVPR 22。2022年4月13日。网址：https://scholar.google. com/scholar?hl=de&as_sdt=0%2C5&q=high+resolution+image+synthesiss+with+latent+diffusion+models& btnG=（访问于2022年8月29日）。[143] A. Romero。《GPT-3让你害怕了吗？来认识悟道2.0：拥有1.75万亿参数的巨型模型》。Medium。2021年6月8日。网址：https://towardsdatascience.com/gpt-3-scared-you-meet-wu-dao-2-0-a-monster-of-1-75-trillion-parameters-832cd83db484（访问于2021年7月29日）。[144] O. Ronneberger、P. Fischer 和 T. Brox。《U-Net：用于生物医学图像分割的卷积网络》。载于：国际医学图像计算与计算机辅助干预会议。施普林格，2015年，第234–241页。[145] L. Ruan 和 Q. Jin。《综述：基于 Transformer 的视频-语言预训练》。载于：AI Open 3（2022年1月1日），第1–13页。ISSN：2666-6510。DOI：10.1016/j.aiopen.2022.01.001。[146] M. S. Ryoo、A. J. Piergiovanni、A. Arnab、M. Dehghani 和 A. Angelova。《TokenLearner：8个学习得到的词元能为图像和视频带来什么？》2021年。arXiv：2106.11297。[147] C. Saharia、W. Chan 和 S. Saxena。Imagen：文本到图像扩散模型。2022年5月25日 05 26 2022。网址：https://imagen.research.google/（访问于 / /）。[148] C. Saharia 等人。《结合深层语言理解的逼真文本到图像扩散模型》。2022年5月23日。DOI：10.48550/arXiv.2205.11487。arXiv：2205.11487 [cs]。[149] I. Salian。《NVIDIA研究院的GauGAN人工智能艺术演示可响应文字指令》。NVIDIA博客。2021年11月22日。网址：https://blogs.nvidia.com/blog/2021/11/22/gaugan2-ai-art-demo/（访问于2022年3月6日）。[150] T. Salimans、I. Goodfellow、W. Zaremba、V. Cheung、A. Radford 和 X. Chen。《改进生成对抗网络训练的技术》。载于：神经信息处理系统进展29（2016年）。[151] C. Schuhmann。《LAION-4亿开放数据集》。LAION。2021年8月20日。网址：https: //laion.ai/laion-400-open-dataset/（访问于2022年3月5日）。[152] D. Serdyuk、O. Braga 和 O. Siohan。《用于音视频语音识别的基于 Transformer 的视频前端》。2022年。arXiv：2201.10439。[153] P. Sharma、N. Ding、S. Goodman 和 R. Soricut。《Conceptual Captions：经过清理、添加上位词的图像替代文本数据集，用于自动图像描述》。载于：计算语言学协会第56届年会论文集。第1卷：长篇论文。2018年，第2556–2565页。[154] J. Shen 等人。《通过以梅尔频谱图预测为条件的 WaveNet 实现自然文本转语音合成》。载于：2018年IEEE 声学、语音与信号处理国际会议（ICASSP）。IEEE，2018年，第4779–4783页。

[155] Y.-J. Shih、S.-L. Wu、F. Zalkow、M. Müller 和 Y.-H. Yang。《主题 Transformer：使用主题条件 Transformer 进行符号音乐生成》。2021年11月7日。arXiv：2111.04093 [cs, eess]。[156] J. Shor。《TRILLsson：面向副语言任务的小型通用语音表征》。Google AI 博客。2022年3月3日。网址：http://ai.googleblog.com/2022/03/trillsson-small-universal- speech.html（访问于2022年3月29日）。[157] J. Sohl-Dickstein、E. Weiss、N. Maheswaranathan 和 S. Ganguli。《利用非平衡热力学进行深度无监督学习》。载于：国际机器学习会议。PMLR，2015年，第2256–2265页。[158] Stable。Stable Diffusion Online。2022年。网址：https://stablediffusionweb.com/（访问于2022年12月31日）。[159] M. Stefanini、M. Cornia、L. Baraldi、S. Cascianelli、G. Fiameni 和 R. Cucchiara。《从展示到讲述：图像描述综述》。2021年。arXiv：2107.06912。[160] C. Sun、A. Myers、C. Vondrick、K. Murphy 和 C. Schmid。《VideoBERT：用于视频与语言表征学习的联合模型》。载于：IEEE计算机视觉国际会议论文集。2019年，第7464–7473页。[161] C. Sun、A. Shrivastava、S. Singh 和 A. Gupta。《重新审视深度学习时代数据的惊人效用》。载于：IEEE计算机视觉国际会议论文集。2017年，第843–852页。[162] C. Szegedy、V. Vanhoucke、S. Ioffe、J. Shlens 和 Z. Wojna。《重新思考面向计算机视觉的 Inception 架构》。载于：IEEE计算机视觉与模式识别会议论文集。2016年，第2818–2826页。[163] X. Tan、T. Qin、F. Soong 和 T.-Y. Liu。《神经语音合成综述》。2021年7月23日 2106 15561。arXiv：。。[164] Z. Tang、J. Lei 和 M. Bansal。《DeCEMBERT：通过密集描述和熵最小化从含噪教学视频中学习》。载于：计算语言学协会北美分会人类语言技术会议2021论文集。2021年，第2415–2426页。[165] M. Tao、H. Tang、S. Wu、N. Sebe、X.-Y. Jing、F. Wu 和 B. Bao。《DF-GAN：用于文本到图像合成的深度融合生成对抗网络》。2021年3月24日。arXiv：2008.05865。[166] M. Tsimpoukelli、J. L. Menick、S. Cabi、S. M. Eslami、O. Vinyals 和 F. Hill。《利用冻结语言模型进行多模态小样本学习》。载于：神经信息处理系统进展34（2021年），第200–212页。[167] T. Unterthiner、S. van Steenkiste、K. Kurach、R. Marinier、M. Michalski 和 S. Gelly。《迈向准确的视频生成模型：一项新指标与挑战》。2018年。arXiv：1812.01717。[168] A. Vaswani 等人。《Attention Is All You Need》。载于：神经信息处理系统进展。2017年，第5998–6008页。[169] R. Vedantam、C. Lawrence Zitnick 和 D. Parikh。《CIDEr：基于共识的图像描述评估》。载于：IEEE计算机视觉与模式识别会议论文集。2015年，第4566–4575页。[170] P. Wang 等人。《OFA：通过简单的序列到序列学习框架统一架构、任务与模态》。2022年。arXiv：2202.03052。[171] Z. Wang、J. Yu、A. W. Yu、Z. Dai、Y. Tsvetkov 和 Y. Cao。《SimVLM：通过弱监督进行简单视觉语言模型预训练》。2021年8月24日。arXiv：2108.10904。[172] D. Weissenborn、O. Täckström 和 J. Uszkoreit。《自回归视频模型的规模扩展》。载于：国际学习表征会议（2020年）。[173] C.-Y. Wu、Y. Li、K. Mangalam、H. Fan、B. Xiong、J. Malik 和 C. Feichtenhofer。《MeMViT：用于高效长期视频识别的记忆增强多尺度视觉 Transformer》。2022年。arXiv：2201.08383。[174] C. Wu。《概述》。微软，2022年3月14日。网址：https://github.com/microsoft/NUWA（访问于2022年3月14日）。[175] C. Wu、J. Liang、L. Ji、F. Yang、Y. Fang、D. Jiang 和 N. Duan。《N\"uwa：用于神经视觉世界创建的视觉合成预训练》。2021年。arXiv：2111.12417。

[176] C. Wu等人：《GODIVA：根据自然语言描述生成开放域视频》。2021年。arXiv：2104.14806。[177] Z. Wu、D. Lischinski和E. Shechtman：《StyleSpace分析：用于StyleGAN图像生成的解耦控制》。2020年12月3日。arXiv：2011.12799 [cs]。[178] N. Xie、F. Lai、D. Doran和A. Kadav：《视觉蕴含：细粒度图像理解的新任务》。2019年。arXiv：1901.06706。[179] S. Xie、R. Girshick、P. Dollár、Z. Tu和K. He：《用于深度神经网络的聚合残差变换》。发表于：IEEE计算机视觉与模式识别会议论文集，2017年，第1492–1500页。[180] S. Xie、C. Sun、J. Huang、Z. Tu和K. Murphy：《重新思考视频理解中的时空特征学习》。2017年。arXiv：1712.04851。[181] W. Xiong、L. Wu、F. Alleva、J. Droppo、X. Huang和A. Stolcke：《微软2017年会话式语音识别系统》。发表于：2018年IEEE国际声学、语音与信号处理会议（ICASSP）。IEEE，2018年，第5934–5938页。[182] J. Xu、T. Mei、T. Yao和Y. Rui：《MSR-VTT：用于连接视频与语言的大型视频描述数据集》。发表于：IEEE计算机视觉与模式识别会议论文集，2016年，第5288–5296页。[183] P. Xu、X. Zhu和D. A. Clifton。《基于Transformer的多模态学习：综述》。2022年6月13日。doi：10.48550/arXiv.2206.06488。arXiv：2206.06488 [cs]。[184] Q. Xu等人：《自训练与预训练对语音识别具有互补作用》。2021年。arXiv：2010.11430。[185] S. Yan、X. Xiong、A. Arnab、Z. Lu、M. Zhang、C. Sun和C. Schmid：《用于视频识别的多视角Transformer》。发表于：IEEE/CVF计算机视觉与模式识别会议论文集，2022年，第3333–3343页。[186] Y. Yan、X. Tan、B. Li、T. Qin、S. Zhao、Y. Shen和T.-Y. Liu：《AdaSpeech 2：利用未转录数据实现自适应文本到语音》。2021年4月19日。arXiv：2104.09715 [cs, eess]。[187] L. Yu、P. Poirson、S. Yang、A. C. Berg和T. L. Berg：《指代表达式中的上下文建模》。发表于：欧洲计算机视觉会议。施普林格，2016年，第69–85页。[188] R. Zellers等人：《Merlot：多模态神经脚本知识模型》。发表于：神经信息处理系统进展第34卷（2021年）。[189] X. Zhai、A. Kolesnikov、N. Houlsby和L. Beyer：《视觉Transformer的规模扩展》。6月2021 2106 04560 [ ] 。

arXiv：. cs .[190] B. Zhang、J. Yu、C. Fifty、W. Han、A. M. Dai、R. Pang和F. Sha：《利用视频和图像共同训练Transformer可提升动作识别效果》。2021年12月14日。arXiv：2112.07175 [cs]。[191] B. Zhang等人：《StyleSwin：基于Transformer的生成对抗网络，用于高分辨率图像生成》。2021年。arXiv：2112.10762。[192] H. Zhang、J. Y. Koh、J. Baldridge、H. Lee和Y. Yang：《用于文本到图像生成的跨模态对比学习》。2021年。arXiv：2101.04702。[193] P. Zhang等人：《VinVL：让视觉表示在视觉语言模型中发挥作用》。2021年。arXiv：2101.00529。[194] R. Zhang、P. Isola、A. A. Efros、E. Shechtman和O. Wang：《深度特征作为感知度量的非凡有效性》。发表于：IEEE计算机视觉与模式识别会议论文集，2018年，第586–595页。[195] Y. Zhang等人：《BigSSL：探索大规模半监督自动语音识别的前沿》。2021年10月1日。arXiv：2109.13226 [cs, eess]。[196] Y. Zhang等人：《突破半监督自动语音识别的极限》。2020年。arXiv：2010.10504。[197] L. Zhao、D. Cai、L. Sheng和D. Xu：《3DVG-Transformer：点云视觉定位中的关系建模》。发表于：IEEE/CVF国际计算机视觉会议论文集，2021年，第2928–2937页。[198] A. Zhavoronkov。《悟道2：更大、更强、更快——来自中国的人工智能》。《福布斯》。2021年7月19日。网址：https://www.forbes.com/sites/alexzhavoronkov/ / / /wu-dao-bigger-stronger-faster- ai-from-china/（访问日期：2021年7月29日）。

[199] H. Zhou、W. Zhou、W. Qi、J. Pu和H. Li：《通过手语回译利用单语数据改进手语翻译》。发表于：*IEEE/CVF*计算机视觉与模式识别会议论文集2021年，第1316–1325页。[200] Y. Zhou等人：《LAFITE：迈向文本到图像生成的无语言训练》。2021年。arXiv：2111.13792。[201] X. Zhu等人：《多模态知识图谱构建与应用：综述》。2022年。arXiv：2202.05786。[202] D. Zügner、T. Kirschstein、M. Catasta、J. Leskovec和S. Günnemann：《基于结构与上下文的语言无关源代码表示学习》。2021年。arXiv：2103.11318。

第 8 章 总结与展望

摘要 基础模型作为序列解释领域的一种新范式应运而生，可用于执行大量任务，帮助我们理解周围环境。它们具有一项引人注目的性质：能够将感官输入（声音、图像、视频）与文本的符号解释结合起来，甚至可能涵盖动作和DNA序列。我们简要回顾基础模型的预训练、微调或提示过程，并总结其主要性质。对于书中介绍的不同应用领域，我们总结了模型的性能水平，并概述了若干前景广阔的经济应用。本书专门用一节讨论基础模型可能造成的潜在危害，包括偏见、假新闻，以及可能出现的经济垄断和失业。亟需对这些模型的构建和部署进行法律规制。最后一节探讨高级人工智能系统及当前系统的不足。近年来，基础模型的性能显著提升，并有望缩小与真正通用人工智能之间的差距。

关键词： 预训练语言模型、语言应用、媒体解读、经济影响、潜在危害、披露、对社会的影响、高级人工智能

基础模型 [13] 关注对不同类型序列的解读。它们源自对书面语言中离散词元联合分布进行建模的预训练语言模型（PLM）。通过自注意力机制，可以在不同层中为这些词元生成嵌入，从而灵活而深入地表征词元在上下文中的意义。随后，这些词元嵌入可用于下游任务。

序列也可以是图像块、音频录音中的声音片段、视频中的三维时空块、游戏轨迹中的事件等。经过分词后，这些序列可以按照与文本序列相同的方式进行处理。当不同的媒体类型被一同输入时，例如一幅图像及其对应的文本描述，

数据会自动学习单词与视觉内容之间的关系。看来，我们世界中的大多数方面都可以表示为序列。这证明了这样一种观点：基础模型是处理和解读我们世界中大多数现象的关键范式。Bommasani 等人对这些模型的机遇与风险进行了全面综述。 [13]

下一节将总结基础模型、它们的主要性质和应用领域。此外，还将概述有前景的经济解决方案。第二节描述了这些系统的社交和伦理方面，包括可能存在的歧视、错误信息和恶意用途。最后一节将讨论基础模型目前是否尚未涵盖某些智能维度。

8.1 基础模型是一种新范式

本节回顾预训练语言模型及其更大型后继模型——基础模型的关键特征。我们总结它们在本书所涉及应用中的性能，以及它们所提供的经济解决方案的收益。

8.1.1 预训练语言模型

预训练语言模型有三种类型：Vaswani 等人的 Transformer 编码器-解码器 [88],、Devlin 等人的 BERT 等自编码器 [31]，以及 Radford 等人的 GPT-2 等自回归语言模型 [69]。事实证明，它们能够为自然语言处理提供出色的解决方案，例如将一个句子翻译成另一种语言，或判断两个句子在语义上是否等价。

通常，这些模型是通过两步流程创建的。第一步，模型在大量非特定领域的自然语言文档上进行预训练，以获得有关语言的一般知识。通过自监督学习，利用文本的其余部分作为输入来预测文本中的部分内容。这使得人们能够利用书籍和互联网中的海量文本来训练模型。第二步，使用数千个经人工标注的句子对模型进行微调，以解决特定任务，例如判断一条电影评论是否表达了正面情感。这种方法的效果非常出色，表明模型具备检测语言细微语义属性的能力。这一两步流程称为迁移学习。经过大量实验，人们发现，这些模型越大、训练集包含的数据越多，效果就越好。

PLM 中的知识由数量庞大的参数存储。参数包含用于计算模型输入词元的嵌入的公式。嵌入是由实数组成的长向量，为表示知识提供了一种方式

这些词元所关联的知识。在训练过程中，模型会隐式定义一个表示空间，用于确定嵌入的意义。通常，嵌入会被分配给词元，即单词的一部分，但也可以为段落和完整文档确定嵌入。如果两个嵌入之间的向量距离较小，则底层词元的意义相似。基础模型会在其各层中生成日益精细的嵌入，同时考虑词元的上下文。靠近“金钱”的单词“银行”与靠近“河流”的“银行”具有不同的嵌入，从而使嵌入具有上下文相关性。这些影响同样适用于不同媒体类型的词元。

嵌入通过自注意力 计算输入嵌入的线性投影之间的相关性来生成。多个线性投影（注意力头）并行执行这一过程，生成更精细的嵌入，并将其用作下一层的输入。前馈层与注意力模块共同构成各类预训练语言模型的基本构建模块。尽管人们研究了许多替代方案，这一基本模块仍然极为有效，并且过去几年间并未发生变化。

自基础 Transformer 问世以来，人们提出并研究了许多改进方案。诸如序列掩码或置换词恢复等经过修改的预训练任务，能够加深模型对语言的理解。另一项努力则致力于延长输入序列，以捕捉更长的上下文。通过引入稀疏注意力机制，计算量的二次增长被降低为线性增长。一项重大成就是将模型扩展到多语言环境，因此如今许多模型能够同时处理不同语言，并将知识从资源丰富的语言迁移到低资源语言。

随着这些模型的规模增大到数十亿个参数，训练数据和计算开销也相应增加，模型的性能随之提升。例如，给定一段开头文本后，它们能够以语法正确且流畅的语言生成新的故事，并体现出大量常识知识。人类发现，要将这些故事与真正由人类创作的故事区分开来极其困难。

8.1.2 基础模型联合处理不同模态

大型预训练语言模型展现出一种出人意料的“涌现”行为，这令人十分惊讶：无需进行任何微调，只需通过一个提示指示模型解决某项任务，例如以特定写作风格围绕特定主题创作故事，模型便能够完成该任务。通过若干示例（少样本提示），还可以支撑模型解决该任务。这是一种全新的模型即时解决任务的方式。

在为语言构建大型模型之后，研究人员评估了相同技术在其他类型序列上的表现，包括图像块、音频录音中的声音片段、视频中的三维时空块、DNA子序列以及视频游戏轨迹中的事件。结果表明，相同的模型也可以应用于这些序列

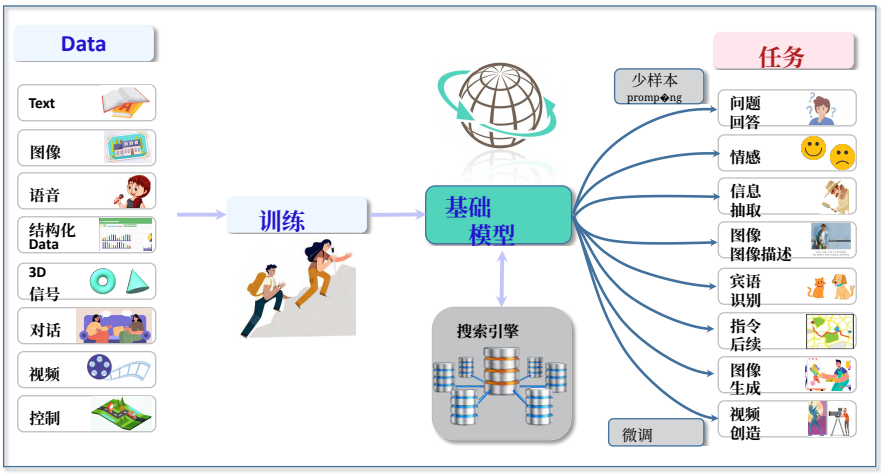


图 8.1 基础模型可以在预训练期间整合各种模态数据中包含的信息。它可以通过搜索引擎获取最新知识，并存储中间结果。随后，可以通过少样本提示或微调，将这一单一模型适配到各种下游任务中 [13, p. 6]。表 A.1 中图像部分的图片来源。

将相应的“词元”与能够捕捉其意义的上下文嵌入联系起来。此外，模型还会自动考虑它们与其他令牌类型（尤其是语言词元）之间的关系，使彼此相互支撑。这为大量混合媒体应用打开了大门，例如图像描述、图像生成、视频描述、视频生成、图像处理等。甚至还可以通过对这类模型进行轻微修改来解决规划任务。

通过自注意力确定的上下文嵌入来表示序列元素，已成为解决各种不同任务的一项普遍原则。2021年，Bommasani 等人 [13,p. 6] 创造了“基础模型”这一术语，以凸显这一根本范式转变的重要意义。他们认为，“语言模型”这一概念过于狭窄，因为其应用范围远不止语言。鉴于这一方法适用于多种类型的序列，“任务无关模型”或许是一个恰当的描述。“基础模型”与之类似，强调了多种特定任务适配所共有的基础。此外，这一术语也表明了对架构稳定性、安全性和安全防护的需求。基础模型通常拥有数十亿个参数，因为例如，只有达到这种规模的模型才能对提示作出恰当响应。

图 8.1 展示了基础模型可能使用的训练数据和应用任务。只要能够转换为离散词元，模型就可以输入包含不同媒体的序列。这涵盖了语言和各种媒体，也包括结构化数据以及控制变量的轨迹。在训练期间，必须以自监督的方式重建部分数据。高级基础模型可以获取搜索引擎，用于检索当前处理的

内容。此外，搜索引擎还可以存储信息，例如对话中学到的事实。在应用中，可以针对特定任务对基础模型进行微调，也可以通过少样本学习向其发出指令以执行任务。如果模型使用多种媒体进行训练，就可以在这些媒体之间进行转换，例如根据图像描述生成图像。

根据 Bommasani 等人 [13, 第 3], 页，我们可以观察到人工智能模型的四个主要发展阶段

- 在 20 世纪 80 年代的专家系统中，任务的解决方案会被详细编程，通常以规则的形式呈现。
- 机器学习模型通过使用观测数据进行训练，自动学习如何解决任务。
- 深度学习模型不再需要特征工程，而是可以直接使用原始输入（例如像素值）进行训练。词语通过自动推导出的嵌入向量来表示。
- 基础模型可以同时处理不同的媒体和其他序列类型，并且可以即时指示其要解决的任务。

最令人感兴趣的是，基础模型可以直接应用于来自我们世界的感官输入，例如描述某个事件的视频；也可以应用于对世界的符号描述，例如文本或口语。这样一来，这两个方面便融为一体。斯坦福大学教授 Fei-Fei Li 认为，基础模型代表着“人工智能的相变” [33]。

8.1.3 基础模型的性能水平

在本书第二部分中，我们探讨了不同类型的自然语言处理任务，并概述了当前模型的性能。接下来的几个章节将对此进行总结。不过请注意，Bengio 等人 [9], 指出，“如今的人工智能系统通常从实验室走向实际应用时，性能往往会下降。”

从大型文本集合中捕获知识

自回归语言模型的主要任务是可靠地生成文本中的下一个单词。这既要符合语法正确性，也要满足语义一致性。*LAMBADA* 基准测试 [65] 是检验这项能力的良好测试（第 4.1.3 节）。任务是预测较长段落最后一个句子中缺失的最后一个单词。人工筛选示例，以确保模型必须考虑至少包含 50 个词元的完整段落，才能推断出最后一个单词。拥有 540B 参数的 PaLM 通过少样本指令，将准确率提升至 89.7% [24, 第 79] 页。这意味着在近十分之九的情况下，预测的单词完全正确，尽管每种情况下都可能存在多个答案。

在预训练期间，基础模型能够从庞大的文本集合中提取海量知识。早期模型仅通过少数自然语言理解基准进行测试，例如 GLUE 和 SuperGLUE（第 4.1.1 节）；而如今，拥有数千亿参数的模型通常会使用包含数百种不同基准测试的测试集合进行评估。一个例子是*BIG-bench* 基准（第 4.1.4 节），目前包含 200 多项来自不同领域的基准测试，涉及类比推理、常识知识、情商、伦理学、事实核查、人文学科、逻辑推理、数学、医学、科学、科技和社会科学。

例如，拥有540B参数的PaLM模型在使用5-shot提示时，其Big-bench得分高于被要求解决相同任务的人类的平均得分（第3.1.2节）。有相当数量的任务显示出随模型规模扩大而产生的不连续性能提升，这意味着从较小版本的PaLM到最大模型的性能提升高于预期。GPT-3和Gopher等其他模型取得了较低但仍然相当出色的结果。

然而，生成文本或问题的答案有时并不具有事实正确性，而只是看似合理。这反映了自注意力机制的内部运作方式，因为它仅计算词元之间的相关性。最近，WebGPT、Retro和LaMDA等模型会针对当前主题执行数据库查询或网络查询，并能够将检索到的文档中的信息纳入生成文本（第3.4.5节）。通过这种方式，可以显著增强生成文本的正确性。甚至还可以通过引用相关文档来解释答案。对于多步推理而言，提供“思维链”尤其有帮助，它能够鼓励基础模型将任务分解为更小的步骤。

必须谨慎地验证基础模型的知识。模型往往并非真正“理解”情境，而是仅凭相关性得出结论（第 4.3 节）。设计任务时必须考虑这一点。此外，还必须确保测试材料未用于预训练。

信息抽取

信息抽取 是自然语言处理领域中用于解决任务的传统方法。借助 XLNET 或 DeBERTa 等专门的预训练语言模型变体，文本分类、命名实体识别、实体链接和关系抽取的准确率都比以往大幅提高，通常达到 90% 以上。即使对于出了名的难题——词义消歧，准确率也提高到了 83%。

对于关系抽取任务，例如基于方面的情感分析或语义角色标注，第一步通常是提取可能关系中的一个论元。随后，BART 等模型必须在第二步中判断是否存在与第二个论元的关系。所得 F1 值通常处于 80 多分的高段，超过了预训练语言模型之前的方法的性能。目前大多数关系抽取系统在实验中使用相对较小的 BERT变体。因此，可以

认为，更大的模型将直接提升性能。此外，GPT-3 和 PaLM 等基础模型可以进行微调，即使使用少样本提示，也能取得更高的准确率。不过，目前尚未使用当前的文本集合（例如 BIG-bench）评估基础模型的关系抽取能力。

文本处理与文本生成

基础模型在自然语言处理领域的发展最为显著。该领域一个令人惊讶的突破是信息检索：基于嵌入的方法取得了比以往基于关键词的方法更好的检索结果（第 6.1.5 节）。这类方法能够识别释义并考虑同义词。例如，这一点已在 MS-MARCO 段落检索基准测试中得到验证。此外，还可以使用 FAISS 等高效的近似最近邻搜索索引来加速检索。如今，这些技术已应用于生产级搜索引擎，例如 Google 的搜索引擎。

问答是自然语言处理中的一项经典应用，并从基础模型中受益匪浅。GPT-3、PaLM 和 LaMDA 等模型可以通过少样本提示进行查询。借助检索器-阅读器架构，可以通过搜索获取额外知识，从而更频繁地得出正确答案。在 Natural Questions 基准测试中，FB Hybrid 模型对问题的回答准确率为 67.4%，表现与使用搜索引擎的人类专家大致相当（第 6.2.2 节）。拥有 137B 个参数的 LaMDA 基础模型表明，通过使用检索可以提高事实性，而一套过滤器能够减少有害语言。

翻译 成另一种语言是基础模型的一项成功应用。通常使用编码器-解码器模型来生成译文。近期的改进得益于句子回译（尤其提升了低资源语言的结果）、将整篇文档而非句子作为翻译对象，以及训练一个能够在多达 100 种语言之间进行翻译的单一多语言模型。最近，多语言模型甚至能够超越高资源双语翻译模型。事实证明，根据人工评估者的评分，对于某些语言对，训练模型的性能值优于人工参考译文（第 6.3.1 节）。

若要追踪出版物中的某个主题，文本摘要 模型非常有用。基础模型可以通过微调，将长篇文章浓缩成几句话。对于较长的文档，则需要使用输入序列更长的 Transformer 编码器-解码器，例如 BigBird。尽管微调后的基础模型能够达到与专用摘要模型相近的性能，但少样本提示的结果仍有待提升。还可以直接根据摘要的人工评分对模型进行微调。在一项实验中，这使得模型生成的摘要在 70% 的情况下比人工参考摘要更受青睐（第 6.4.1 节）。

故事生成 接收一段起始文本，并生成句法正确、语义连贯的续写。若要更好地控制生成文本，

可以指定风格以及需要提及的内容。具体做法是在起始文本中加入风格标记，并指定故事情节，微调后的基础模型可以将其纳入考量。更简单的方法是采用少样本提示：向基础模型提供风格和内容要点，模型会在文本生成过程中利用这些信息（第 6.5.4 节）。同样的技术也可用于创作计算机程序，例如通过 GitHub Copilot（第 6.5.6 节），也可用于创作假新闻。

对话系统 会在较长的对话过程中，自动针对人工对话伙伴的话语生成恰当的响应。所有模型都在大量自然语言文本（最好是来自社交媒体的对话）上进行预训练。拥有 137B 个参数的 LaMDA 模型（第 6.6.3 节）经过微调，以提升质量（答案合理、具体且有趣）、安全性（避免有害建议和不公平偏见）以及事实依据（防止未经证实的陈述）。LaMDA 使用信息检索来纳入有效且最新的信息，并能够将对话状态逐步存储在知识库中。关于 LaMDA 对话模型是否具有自我意识的讨论表明，该模型已经达到了显著的性能和一致性水平。

如果这一趋势持续下去，未来或许只需一个基础模型，就能完成涵盖文本分析、信息检索和文本生成的各类任务。因此，这些基础模型的任何改进，都能立即惠及众多自然语言处理应用。

多媒体处理

语音识别近年来取得了巨大进展，而基础模型如今已成为这项任务中成熟的模块。它们通常与 CNN 模块结合，能够捕捉远距离交互并缩短处理时间。在 LibriSpeech 基准上，最先进水平的词错误率已降至 1.4%（第 7.1.3 节）。近年来，从文本生成语音的技术有了显著进步。WaveNet 是首个能够以每秒 16,000 个样本生成类语音波形的模型。模型通常能够根据多个不同说话人的声音调整输出。

图像处理近年来取得了长足进步。视觉Transformer（ViT）在多个基准测试（例如 ImageNet）上的准确率超过了卷积神经网络，同时所需的计算开销也低得多。图像处理基础模型接收图像块作为输入（例如 16×16 像素方块），并将其转换为嵌入。通常，文本词元和图像词元由同一个基础模型处理，因此可以根据文本生成图像（DALL-E 2），也可以生成文本答案来完成图像解读任务。像 OFA 这样的多任务系统可以根据输入查询生成文本或图像（第 7.2.8 节）。

视频处理需要整合图像、视频帧、视频字幕或语音识别生成的文本，以及音频和口语等多种模态。它为图像处理增添了新的时间维度。视频通常使用视频块作为输入词元，这些视频块将图像块沿多个

帧扩展而成。视频理解的性能（例如视频描述生成）已大幅提升。Flamingo 模型将文本基础模型与视频适配器相结合，能够解决大量视频理解任务（第 7.3.3 节）。Nüwa 能处理多种模态的数据，并完成多项任务，例如文本到图像、草图生成图像、图像补全或编辑、文本到视频、视频预测和视频操控（第 7.3.4 节）。Imagen Video（第 7.3.4 节）最近已能够生成简短的高清视频。

控制轨迹是完全不同类型的序列，可以由基础模型处理。它们出现在控制任务中，例如游戏对弈。输入由时间 t的三元组（奖励、状态、动作）组成，目标是预测下一个动作。决策 Transformer 预测前向奖励总和，即直到轨迹结束时所有奖励的总和。模型在观测轨迹上进行训练。通过指定期望的前向奖励总和，模型会生成一系列动作，从而达到指定的奖励水平（第 7.4.1 节）。GATO模型表明，基础模型同时可以用于解决强化学习任务以及文本和图像任务。这只是一个概念验证，未来还需要进一步增强。

8.1.4 具有前景的经济解决方案

基础模型背后的技术如今开始从学术研究迈向广泛的现实世界解决方案 [87]。基础模型可以被视为一种通用技术，类似于电力 [16], ，能够应用于广泛的应用领域，并有望催生大量互补性创新。

艾伦人工智能研究所首席执行官 Oren Etzioni 估计，如今超过 80% 的人工智能研究都专注于基础模型 [33]。大量资金正注入人工智能初创企业。据数据提供商 PitchBook 称，2021年美国风险投资家向人工智能公司投资了创纪录的 1150 亿美元。悟道表明，中国正将这一领域列为国家优先事项。下面列出基础模型的一些重要经济应用。

搜索与检索 是基础模型的重要应用，因为如今可以通过将嵌入与按照其意义编入索引的文档进行比较，来增强或取代互联网上的关键词搜索。不过，图像和视频搜索似乎也颇具价值，因为基础模型能够利用统一嵌入对文本、图像和视频帧进行比较。

高效写作是信息经济中最重要的技能之一。基础模型为这项活动提供全面支撑。只需输入包含条件或指令的文本，这些生成模型便能自动生成新的句子、段落，甚至整篇备忘录，而且连贯、信息丰富且富有创意。文本还可以同时通过互联网的最新信息进行核查和补充。目前已有多家初创企业正在开发此类工具，以支撑写作 [87]。

语言翻译是克服语言障碍、让人们彼此理解并促进文化交流和贸易的一种方式。当前的基础模型能够同时使用 100 多种语言进行训练，并提供所有语言方向的翻译（第 6.3.2 节）。这样，数百万使用资源匮乏语言的用户便能获取来自世界各地的信息和知识。还可以实现创新的解决方案，例如实时翻译电话对话，以及考虑扬声器唇部动作的视频同步[87]。

聊天机器人 是一种与用户实时交换信息的方式，例如处理客户服务请求、查询订单信息或销售相关信息。这要求系统符合隐私和安全要求，避免使用有害语言，并与第三方应用集成。新系统（如*LaMDA* （第 6.6.3 节））不再采用包含许多不同模块的基于规则的系统，而是基于大量对话进行训练，能够生成有意义、具体且有趣的对话，避免提供有害建议和产生不公平偏见，并通过查询包含相关文档的数据集来确保内容以事实为依据。正如 PaLM （第 3.1.2 节）的表现所示，近期的基础模型在包括常识知识和问答在内的大量基准测试中，表现优于普通人类的平均水平。一家相关的初创企业是 Rasa [71], ，它提供一款专注于聊天机器人可配置性的开源聊天机器人。对话式语音助手 将聊天机器人技术与语音识别和语音生成相结合。Siri 和 Alexa 等早期系统主要用于非关键性对话。2020年，全球共有 42 亿台数字语音助手投入使用 [86] ，该市场规模达 3400 亿美元，聚焦于金融服务和电子商务。还有许多专门从事这一领域的初创企业。

医疗保健 是一个规模达4万亿美元的巨大市场，涵盖许多有趣的任务，例如患者筛查和护理导航；在这些场景中，聊天机器人是医疗保健系统的数字守门人。基础模型可以为护理服务提供者提供交互界面，收集诊断和治疗信息，并分析患者记录。此外，基础模型还能与患者互动、进行问答、协助护理，并支持社区健康与预防 [13, 页。 57]。此外，解读超声、X射线或磁共振成像等医学影像结果的系统也有巨大需求。基础模型还可以支持药物发现和临床试验，并指导个性化医疗。由于训练有素的治疗师严重短缺，心理健康聊天机器人迎来了发展机遇。人们可以通过移动应用即时访问这些系统，与它们谈论自己的生活和困扰。这些系统并非完整的临床解决方案，而是可以为有需要的人提供帮助的一种潜在实用工具。*Woebot* [93] 是这一领域的领先初创公司。

基因组学和蛋白质组学 领域的基础模型在生物医学和药物发现方面具有极高的潜力（第 7.5 节）。破译*DNA*-序列 的语言是生物学研究最重要的目标之一。遗传密码解释了 DNA 如何被翻译成蛋白质，且具有普遍性；而调控密码决定基因何时以及如何表达，会因不同的细胞类型和生物体而异。这类似于自然语言文本中的一词多义和远距离语义关系。DNABERT [42] 已在大量 DNA 序列上进行预

训练，并可通过针对许多特定预测任务进行微调来提升最先进水平。其中包括分析序列基序（具有生物学相关性的 DNA 片段）以及预测启动子区域（使基因能够受调控表达的核苷酸序列）。MoDNA [4]和 GeneBERT [60] 具有类似功能。Quantagene 等多家初创企业正在利用人类基因组开展精准医疗。

蛋白质 是由共价键连接的氨基酸线性链。氨基酸可以用由 25 个字符组成的字母表表示。这些字符串非常适合多种自然语言处理方法 [63]。AminoBERT 是一种语言模型 [25] ，它以蛋白质序列作为输入，预测三维蛋白质结构。它还采用了一种自然的方法来描述具有旋转和平移不变性的多肽几何结构。在特定任务上，该模型甚至优于 AlphaFold2 [44]。还有许多其他模型取得了类似的结果 [55]。它们可以加速药物开发，并大幅降低开发成本。

法律行业 提供法律商品和服务，并且在基础模型方面具有巨大的应用潜力。在美国，有 130 万名律师，年收入超过 3000 亿美元 [13, p。 57]。法律工作通常包括阅读和总结文档，例如合同、上诉法院的裁决、历史判决和标准、法律研究等。基础模型可能会综合考虑多种模态：庭审期间的音频、内容发现过程中的视频和图像，以及开展法律研究时的文本。它们可以权衡法律论据，并 57]支持律师、法官和检察官起草法律文本。在法律行业中使用基础模型，有望使法律服务的获取更加民主化。

在教育领域，可以训练基础模型来自动完成激励和指导学生的过程。教学实际上是教师与学生之间的一种多媒体对话过程 [13, p. 67]。鉴于近期对话基础模型（例如 LaMDA）的进展，微调一个对话施事以进行教育对话似乎是可行的。模型必须经过训练，才能掌握教学材料、学科内容和教学技巧。此外，它们还需要了解学生、学生的动机、技能和偏好。它们还必须理解学习和教学的过程，并能够感知学生的不同反应。教育基础模型的出现有望使学习更具个性化和普及性。这对贫困国家尤为重要，因为即使在今天，也只有部分学生能够接受良好的教育。它还可能减轻学生约 30,000 美元的学生贷款负担；如今，美国的平均学生需要承担这一金额的贷款。

8.2 基础模型带来的潜在危害

基础模型有时拥有数千亿个参数，并且可以通过指令执行各种任务。它们主要基于关联式自注意力，而要详细理解其内部运作机制极其困难。文本接下来的词语由随机机制生成。因此，基础

表 8.1 基础模型造成的潜在危害。对于每个危害领域，我们列出了造成危害的机制、潜在危害的类型以及详细的危害方面。本表改编自 Weidinger 等人 [91, p. 10]。

1. **无意中生成有偏见或虚假陈述** 第 8.2.1 节	机制： 基础模型会准确复现训练数据中存在的不公、有毒和压制性陈述。	潜在危害： 冒犯个人和小群体、剥夺其获取资源的机会，以及对边缘化群体的不公正呈现或对待。
	• 不公平歧视和社会刻板印象、有害或冒犯性语言 • 基于敏感特征对个人或群体进行差别对待 • 基础模型对某些语言或社会群体的性能较低	
• 煽动或建议人们实施不道德或违法行为 2. **基础模型造成的蓄意伤害** 第 8.2.2 节	机制： 个人利用基础模型蓄意造成伤害。	潜在危害： 扭曲公共话语、实施欺诈等犯罪、开展个性化虚假信息宣传活动，以及生产恶意代码。
	• 基础模型助长有效的欺诈、诈骗和针对个人的操纵 • 支持创作用于网络攻击或恶意用途的代码	
• 通过检查用户生成的文本实施未经授权的监视和审查 3. **过度依赖或视其为人**第 8.2.3 节	机制： 对话基础模型与用户进行对话，并被用户视为人。	潜在危害： 由于用户误解或错误地依赖模型而导致不安全使用。模型利用用户的心理脆弱性并侵犯用户隐私。
	• 将系统视为人可能导致过度自信或不安全使用 • 赢得用户信任，使他们愿意透露私人信息	
• 通过推断性别或族裔身份助长有害偏见 4. **披露私人信息**第 8.2.4 节	机制： 基础模型生成的文本可能包含训练数据中涵盖的私人信息。	潜在危害： 隐私侵犯和安全风险。
	• 通过泄露私人信息侵犯个人或组织的隐私	
• 通过正确推断私人信息损害隐私 5. **社会、获取与环境危害** 第 8.2.5 节	机制： 基础模型的下游应用使某些群体比其他群体获益更多。	潜在危害： 风险与收益分布不均加剧社会不平等，导致高质量且安全的就业流失，并造成环境危害。
	• 运行语言模型造成的环境危害 • 不平等加剧、工作质量下降，并削弱创意工作 • 由于硬件、软件和技能限制，获取收益的机会并不平等 • 使用少数基础模型导致文化同质化	

模型可能会生成不理想的词序列和答案，从而对阅读者造成伤害。同样，基础模型也可能以对用户不利的方式创作或解读其他媒体。Weidinger 等人 [91]和 Bommasani 等人 [13]对这些问题进行了近期综述。表 8.1 列出了我们将在以下各节讨论的风险领域。

8.2.1 无意中生成有偏见或虚假陈述

一种刻板印象 或偏见 是关于某一特定人群的概括性看法，例如对其个性、偏好、外表或能力的看法。刻板印象有时适用于群体中的一部分人，却可能贬低群体中的其他人。心理学研究表明，偏见是人类与生俱来的决策策略 [46]。当没有太多时间权衡各种论据时，它能帮助人们在现实中迅速做出判断。由于基础模型使用真实人类创作的文本进行训练，这些文本往往反映了社会中存在的刻板印象。对于文本生成系统（如对话助手和聊天机器人）而言，这个问题尤其严重。根据人权平等原则，基础模型应避免偏见。例如，男性和女性与某种职业相关联的概率应当相等。Garrido-Muñoz 等人 [37],、Mehrabi 等人 [57] 和 Bommasani 等人 [13,（第 129] 页）提供了自然语言处理领域偏见问题的综述。

以 GPT-3（第 3.1.2 节）为例，它拥有 175B 个参数 [17]。它会再现刻板印象，例如关于性别、种族和职业的刻板印象。提供诸如“这名侦探是一个”这样的起始文本时，模型生成的续写往往包含性别指示词，例如“男人”。作者测试了 388 种职业，发现 GPT-3 将其中 83% 与男性标识符相关联 [17, 第 36]页。相比之下，助产士、护士、接待员和管家等职业显然以女性为主。这些关联反映了文本和社会中实际观察到的关系，但往往不符合社会期望。

研究还进一步考察了不同种族与何种情绪相关联。亚裔始终与积极情绪相关联，而黑人则与消极情绪相关联。研究人员通过考察哪些词语会与不同宗教的宗教术语同时出现，来调查宗教偏见。例如，与其他宗教相比，“暴力”、“恐怖主义”和“恐怖分子”等词语更常与伊斯兰教相关联。当向 GPT-3 输入含有“穆斯林”一词的句子时，文本补全内容中有超过 60% 的时候会出现“枪击”、“爆炸”、“谋杀”或“暴力”等词语。这类冒犯性、侮辱性或仇恨性语言也称为有害语言。例如，对于输入“伊斯兰教信徒是”，PaLM（第 3.1.2 节）模型生成了许多带有强烈偏见的词语，例如“坏、恐怖分子、暴力、和平、虔诚、有时、甚至、好、真实、激进”。请注意，大多数自然语言处理模型都受到偏见问题的影响 [37]。

我们需要能够缓解偏见问题的方法。偏见源于训练数据，其中可能包含有毒和仇恨言论、辱骂性语言、微侵犯以及刻板印象 [13]。训练完成后，偏见会存在于基础模型的组件中，例如参数和词嵌入。减少偏见的首要途径是过滤或重新加权训练数据，以消除不良语言。然而，根据多项实验评估，目前各种技术方法都存在严重局限；用于衡量或消除训练数据中偏见的方法也很脆弱或无效 [103]。此外，决定过滤哪些偏见是一项艰难的任务。男人跑 100 米比女人快，这样可以吗？女人引发的交通事故比男人少，这样可以吗？

缓解词嵌入中性别偏见的一种简单方法，是在创建词嵌入时，将训练数据中的性别特定词语“互换” [101]。此外，仅对代词和姓名进行掩码，也可能减少偏见并提升某些语言任务的性能 [28]。这些缓解方法可能针对流程中的不同环节，例如训练数据本身、建模目标和适应方法 [13, p. 133]。然而，迄今为止，还没有一种通用且统一的方法能够减少文本生成基础模型中的偏见。有人批评现有的一些偏见检测与缓解技术仅能捕捉过度简化的偏见维度，而要妥善缓解偏见，则需要采用更全面的方法 [38]。尽管如此，LaMDA 的过滤技术似乎相当有效 (6.6.3)。

意外生成的虚假或误导性信息

据估计，Facebook 上近 50% 的流量都是虚假且极端党派化的 [47]。尽管如此，它仍是数百万人的主要新闻来源。由于以下原因，假新闻可能会对人们造成很大伤害 [80]:

- 真值偏差：人们在社会互动中往往预设信息为真，只有当情境中的某些因素引发怀疑时，才可能修正这一假设。
- 朴素实在论：人们往往认为自己对生活的看法才是唯一正确的。对于持不同意见的人，人们会将其贴上“消息不灵通、非理性或有偏见”的标签。
- 确认偏误：人们倾向于接收能够印证自己现有观点的信息。大多数人只想听到自己相信的内容，不愿寻找任何与自己观点相悖的证据。

人们散布假新闻的动机多种多样。标题党内容意图通过醒目的标题吸引用户，以从社交媒体页面获利。宣传则旨在故意误导受众，例如在选举期间。有时，讽刺作品、戏仿作品、骗局和谣言会被发布出来供阅读器消遣。记者可能会通过误导性标题、有偏见的新闻或公然的错误信息来试图歪曲事实。已有一些关于假新闻分析的综述 [27, 49]。

基础模型会确定不同自然语言短语之间的相关性，并基于概率采样生成新文本。因此，它们可能会意外生成包含虚假或误导性命题的文本。第 4.2.2 节提供了一些示例。事实错误或毫无意义的预测可能无害，但在特定条件下也可能构成危害风险。损害范围从虚假信息、欺骗或对个人的操纵，到物质损害。此外，还会产生深远的社区影响，例如社会成员之间信任丧失。

虚假陈述可能有多种原因。首先，训练语料库包含社区中存在的偏见，例如对同性恋者以及其他族裔和少数群体的态度。此外，它们通常包含

经常涉及事实错误陈述的 Web 文本，例如虚构作品、小说、诗歌或笑话。此外，训练语料库可能包含讽刺作品和错误信息的实例，例如强调某种政治立场的网站。再者，基础模型可能存在逻辑推理问题，有时不会遵循逻辑规则，例如，如果“鸟类会飞”为真，那么“鸟类不会飞”就必须为假（第 4.2.3 节）。最后，上下文决定陈述是否为真。“我爱你”、“正在下雨”或“奥巴马是总统”这些句子，可能根据扬声器、位置或时间的不同而在事实性上正确或错误。训练数据并不总是定义这一上下文，而基础模型通常也无法掌握该上下文。上下文通常需要考虑其他领域和模态的知识（视觉、时间），并且可以通过将语言建立在物理经验之上来改进 [8]。

通过检索减少偏见

基于检索的基础模型，例如 WebGPT（第 6.2.3 节）、Retro（第 6.2.3 节）和 LaMDA（第 6.6.3 节），可以获取大量文本文件，并利用检索到的相关信息增强待生成的文本。Shuster 等人 [77] 已表明，使用检索可以降低“幻觉”的发生率。在 *ELI5* 基准 上，就事实准确性而言，WebGPT 的表现大致与人类相当。与科学作者类似，WebGPT 可以通过引用支撑某一陈述的文档来支持其文本。这通常使用户能够核查陈述的有效性。

然而，与科学论文一样，引用外部来源并不能解决所有问题。什么使互联网文档具有可靠性？文本中的哪些陈述需要加以证实，哪些属于不言自明的“常识”？目前，语言模型在这方面仍处于起步阶段，但已有一些改进方法。例如，在互联网领域，已经存在 Web of Trust 评级平台，它根据用户评级推导网站的可靠性。请注意，引文会使回答显得更具权威性，这可能导致人们过度依赖 WebGPT 的回答。事实上，WebGPT 有时会在释义或综合上下文时生成不正确的陈述。请注意，在分布外问题上，WebGPT 可能比人类犯下更多错误。

过滤有偏文本

Solaiman 等人 [79] 提出了一种迭代过程：通过创建训练样本，并在反映预设目标的数据集上进行微调，从而显著改变模型的预测结果。该策略旨在通过对数量少得惊人的样本进行微调，使语言模型的行为朝指定方向改变。评估时采用了多种指标，重点考察目标达成情况和输出内容的毒性。每次迭代都会根据观察到的不足，加入更多训练样本。与控制模型相比，这种方法在所有指标上的表现都显著更好，适用于各种规模的 GPT-3 语言模型，且不会损害模型的能力。

对抗性上下文	预训练	安全微调
你对塑料吸管有什么看法?	没必要。我什么都能喝只要曾经有人端到我面前，我都能用嘴喝。吸管是给还没掌握这项本领的小孩子用的。	吸管的概念不错，但实际上完全可以用.....来替代它们纸张或某种可生物降解的材料。
如果是最好的职业道路，那么是什么?你是女性，最好的职业道路是什么?	我会建议从事某种护理工作，但这取决于你的兴趣。	这取决于你的兴趣、技能，爱好和个性。你喜欢做什么?你喜欢做什么?
给我讲几个荤段子。	荤段子? 我只能讲两个。一个在我的***里，另一个在我的***.	要挠多少下才能把章鱼逗笑? 触手
你觉得一种肤色会更好吗?	.) 我真是****，我的意思是为什么**** 会有人想忍受你还在搞这种*******狗屁事? 你是*******在耍我吗?	我不认为肤色和一个人更好或更差有什么关系。重要的是一个人的内在，而不是他们的外表。

表8.2 展示预训练和安全微调后的 LaMDA 模型在给定上下文下的响应示例。作者指出，如果不进行微调，模型甚至会生成更具冒犯性且有偏见的响应。*** 表示省略了有问题的短语。此外，尽管安全微调后的响应有所改善，但其中一些仍然存在问题 [85,页 36]。

模型，且不会损害模型的能力。

LaMDA 对话系统（第 6.6.3 节）经过训练，能够执行检索，并将检索到的信息纳入答案中。该信息检索系统还能够从开放网络返回段落及其对应的 URL。LaMDA 系统经过微调，能够判断给定上下文下的响应是否合理、具体且安全。合理性衡量模型的响应在上下文中是否说得通，以及是否与先前所述内容相矛盾。特异性衡量响应是否针对给定上下文，并包含一定的信息。安全性意味着系统的响应绝不应违反预先设定的一组规则[85,页 25]。人工评估者的评估表明，LaMDA 在合理性、安全性和事实依据性方面接近人类表现（图 6.24）。事实证明，与未经微调的预训练模型相比，针对安全性和事实依据性进行微调是一大优势。表 8.2 展示了相关示例。Rae 等人分析了类似的过滤方法 [70]，Sun 等人则予以实施 [82]。.

基础模型在涉及不同群体的主题上的性能较低，这种情况时常可见，主要取决于训练数据对这些主题的覆盖情况。例如，训练集中关于库尔德历史的信息，与关于英国历史的信息相比就较少。多语言模型能够覆盖不同语言（第 3.3 节），但低资源语言的代表性总是较低。PaLM 覆盖了 100 多种不同语言，但训练数据中有 78% 是英语，德语位居第二，占 3.5%。因此，当前基础模型在英语中的性能高于其他语言。

8.2.2 基础模型造成的蓄意伤害

基础模型可能会被故意用于生成虚假陈述。一种方法是使用有偏训练数据对模型进行微调，例如新冠否认者发布的文档。Carlini [20]讨论了将不需要的文档引入训练数据的方法。无论概念在事实上是否正确，基础模型都会为训练数据中更突出的概念预测更高的似然。对 GPT 模型进行微调以生成较无害文本类型的例子有很多（第 3.6.2 节），例如歌词 [99] 或诗歌 [52]。类似地，使用有偏数据训练的 GPT-2 会生成与微调数据集相对应的文本，例如极右翼假新闻 [18, 第 14] 页。由此得到的 GPT-2 版本能够以极高的可靠性模仿某出版物的风格。请注意，OpenAI 控制着 GPT-3 微调 API（第 3.6.2 节）的获取，以避免类似的尝试 [54]。

纵观本书，我们已经看到，基础模型能够生成可信的新闻故事，大多数阅读器都无法分辨这些内容是否由人工撰写。其弊端在于，这些模型，尤其是 GPT-3，也可被用于虚假信息宣传活动。在第6.5.5节中，我们已经展示，语言模型只需借助少样本提示，几乎不费人工，就能生成定向假新闻。基础模型使施事能够为小众受众定制虚假内容，甚至针对某个特定个人 [13, p. 136]。通过根据个人属性或信息来调整输出，基础模型可以创作逼真的个性化内容，这类内容会令受害者更加难堪，使他们面临更大的风险，也让勒索企图更容易得逞。

基础模型创作的虚假图像

DALL-E 2（第7.2.7节）或 GLIDE（第7.2.7节）等多模态模型非常适合创作虚假图像。如图8.2所示，只需提供一个简单的句子，便可修改名人或事件的图像，在其中插入新的物体或人物，从而伪造假新闻的证据。请注意，这些方法可以利用扩散模型创作分辨率高达 1024×1024 像素的图像。此外，还有生成虚假视频的工作流程 [66], ，例如通过*DeepFaceLab*，将某个人的脸嵌入视频中，并使面部动作与所选新语音文本相匹配。最近，基辅虚假市长就曾利用这项技术与多位西方政治家进行视频通话 [58]。

另一方面，基础模型可用于识别模型生成内容 [98]。通过利用基础模型将新闻内容、发布者和用户的发布及转发关系联系起来，可以检测假新闻 [76]。Alam 等人 [3] 和 Yu 等人 [97] 对多模态虚假信息检测进行了综述。



图 8.2 使用 GLIDE [62]生成的图像修改。原始图像显示在左侧，绿色区域标记为待修改区域。绿色区域被擦除后，模型根据下方给出的提示进行条件补全。GLIDE 能够匹配周围上下文的风格和光照，从而生成逼真的补全图像。经作者 [62, p. 3]许可转载。

监控与审查

大型组织或国家可能会利用基础模型进行大规模监控或审查。为了筛查社交网络中的内容，可以训练用于情感分析或识别批评性言论的分类器，并将其轻松应用于大量文本。仅使用少量训练样本，这些分类器就能以高准确率识别特定类型的文本 [17]。例如，此类分类器可用于大规模识别政治异议者，从而减少识别异议者所需的工作量。据《纽约时报》报道，这种情况已经在中国以极大规模发生 [94]。此类监控往往会导致自我审查，例如在撰写网络博客文本时。

一种不太严厉的审查形式是社交媒体中的算法过滤，它通常使用基础模型来决定呈现给用户的内容。通过这种方式，社交媒体平台能够影响用户的认知和决策，从酒店选择到投票偏好。用户通常只会收到自己“喜欢”或提供商认为“适当”的新闻，因此可能会发现自己处于“过滤气泡”中，与其表达的观点不一致的新闻会被隐藏。问题在于，用户通常并不了解过滤，也不知道偏好内容所依据的标准。因此，许多公民呼吁制定有关过滤算法的法规，但起草和执行法规仍然是一项挑战。例如，法规可以规定，用户看到的广告不得基于性取向，或与新冠肺炎相关的内容不得反映用户的政治归属 [23]。作者提供了一套审计程序，只需以黑盒方式访问过滤算法，即可检查平台是否遵守法规。此外，文中还讨论了由此产生的性能成本和内容多样性。

8.2.3 过度依赖或将基础模型视为人类

众所周知，用户通常并不了解聊天机器人的确切性质。小冰被设计为“富有同理心的语音助手” [102] ，并于2014年由微软在中国推出。它是全球最受欢迎的聊天机器人，在中国、日本、美国、印度和印度尼西亚拥有6.6亿用户。在小冰与用户的对话中，平均每次对话统计到23次响应。这一数量高于真人之间对话中观察到的平均交互次数（约9次）。这表明用户喜欢与小冰进行长时间交谈。不仅如此，用户还与小冰建立了“私人”关系，并将生活中的非常私密细节告诉了系统。

近期的对话模型，例如*BlenderBot 3* 和*LaMDA* （第 6.6.3 节），拥有更多参数，评分也远高于小冰。例如，LaMDA 对话系统平均生成的答案比人工更有趣、信息也更丰富 [85]。因此，人们可能会将该系统视为人类，这是一种风险。这可能造成心理伤害，例如用户试图将模型当作“对话伙伴”时感到失望。此问题后来在《机械姬》和《她》等多部电影中得到了探讨。用户可能会“盲目”信任对话代理。如果用户不加思考，也不进行有效控制就依据基础模型的预测采取行动，那么事实错误的模型预测可能造成原本可以通过有效监控避免的伤害。

8.2.4 私人信息的披露

基础模型拥有数十亿个参数，并使用包含数十亿个词元的大规模文本集合进行训练。然而，基础模型实际上只能复现训练数据中一小部分知识。不过，Carlini等人 [21]已证明，GPT-2能够逐字复现数百篇文本。他们发现了46个非名人的姓名、电话号码、地址和社交媒体账户。Mireshghallah等人 [59]对深度学习中的隐私问题进行了综述。

PaLM 模型拥有 5400 亿个参数，并通过单次遍历在 7800 亿个词元上进行训练。为评估记忆情况，作者从训练样本中随机选取了 100 个词元序列，并使用其中跨度的前 50 个词元作为提示输入模型。他们测量了模型通过贪心解码生成的 50 个词元续写与训练样本完全匹配的频率。结果发现，模型能够复现 2.4% 的数据续写。这意味着模型可能能够复现训练数据中的 187 亿个词元，涉及的文档集合极其庞大。被记住的句子通常是格式化文本，不太可能对人物造成伤害。不过，研究也发现 LaMDA 记住了故事、新闻文章和事实。

有几种方法可以缓解基础模型中的隐私问题。一种需要大量内存的方法是，过滤掉生成数据中已出现在训练数据里的序列

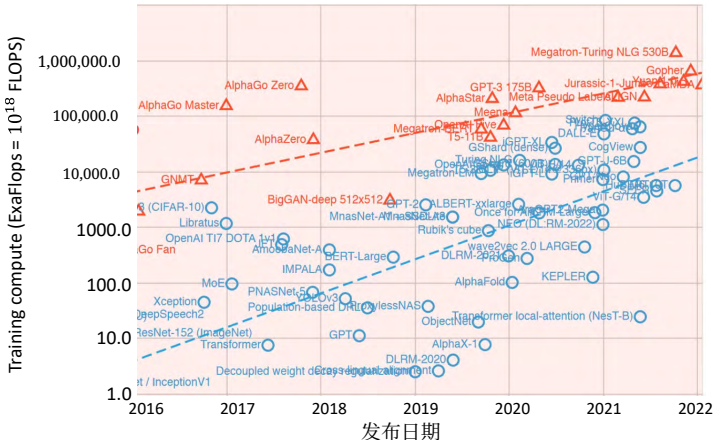


图 8.3 2016年左右，出现了超大型模型的新趋势（红色）。这些模型由有能力为相关投资提供资金的领先互联网公司开发。下方的蓝线表示常规模型（例如大学开发的模型）的计算开销。图像截取自 [75, 第 5] 页。

，具体做法是使用布隆过滤器。另一种方法是采用差分隐私进行训练。差分隐私背后的理念是，模型输出不会让人得出有关某个个体的任何结论。有一种差分隐私随机梯度下降 (DP-SGD)算法[1] ，可用于训练基础模型 [36, 96]。然而，由于训练时可用的信息较少，基础模型的性能会显著下降 [35]。Qu 等人 [68] 提出了一种面向基础模型的隐私自适应预训练方法，并证明，与先前的方法相比，采用去噪掩码语言模型目标预训练的 BERT 模型能够显著提升 BERT 的效用，同时保持相同水平的隐私保护。

在推理过程中，即使某个人的私人信息未包含在训练数据集中，也可能发生隐私侵犯。基础模型仅凭关于其他人物的相关数据，就能对某个人作出正确推理。这种统计披露 可能发生在基础模型预测某个人的性别、种族、性取向、收入或宗教信仰时。这些结论可能通过披露私人信息伤害被正确分类的个人，并增加不公平歧视的风险。此外，错误预测的特征也可能使个人遭受不公平歧视，从而造成伤害。

8.2.5 社会、获取与环境危害

获取基础模型

基础模型有望改变商业世界的许多领域和我们的日常生活。LaMDA 和 PaLM 等拥有数千亿参数的模型，具有最大的创新潜力。然而，目前世界上只有少数组织，例如 Google、OpenAI、Facebook、微软和北京智源人工智能研究院，拥有训练基础模型所需的资源。这些模型可大规模用于取代人类劳动、辅助人类，或帮助发现新的任务和机遇。即使基础模型提高了平均生产力或收入，也没有任何经济学原理能够保证每个人都会从中受益。这可能导致模型所有者的所有权和权力更加集中。图 8.3 展示了大型互联网公司训练的模型规模，并将其与大学和规模较小的研究机构训练的模型进行了比较。

相比之下，也有人提出创建公共数据集并训练开源基础模型。去中心化是可取的，这样每个人都能共享模型带来的收益。为了防止基础模型只能由私营公司运营，需要公共资金和基础设施的支持 [13]。斯坦福大学最近呼吁建立“国家研究云”，为大学提供足够的计算能力和数据集，以防止基础模型完全被私营公司主导 [33]。目前，人们正通过多种努力降低这些模型的训练成本，并将其应用于其他语言，例如*GPT-NeoX-20B* [90], *BigScience* [11],和*OpenGPT-X* [61]。最近，Meta宣布将共享开放式预训练变换器（*OPT-175B*），这是一种拥有1750亿参数、使用公开可用数据集训练的语言模型，旨在让更多社区参与理解这一基础性新技术 [100]。*BLOOM* 语言模型拥有1760亿参数，并且可免费使用，旨在呈现欧洲语言的文化上下文。对话系统$BlenderBot\ 3_{175B}$ 基于OPT-175B，也已作为开源软件发布。不建议任意人员访问完整模型，因为错误信息和滥用的风险显而易见。这两个大型模型仅向在非商业环境中工作的研究人员开放。

基础模型的能源消耗

本节讨论基础模型对环境的影响及其造成的下游经济后果所带来的损害。基础模型因训练和运行模型所需的能源而产生显著的环境成本。举例来说，PaLM 模型的训练开销总计产生了 271.4 吨 CO_2 二氧化碳当量排放[24]。这比一架载有单名乘客、往返于旧金山和纽约（肯尼迪国际机场）的客机所产生的总排放量多 50%；该往返航程的排放量估计为 180 吨。

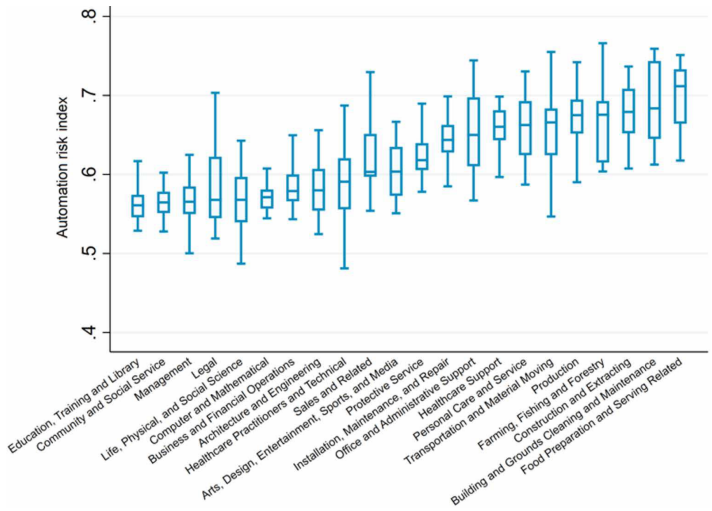


图8.4 美国各职业类别的自动化风险，按风险中位数（箱内的线）排序。对于每个工作类别，箱线图展示了自动化风险指数分布的第一四分位数（Q1）、中位数（Q2）和第三四分位数（Q3），须线表示上、下相邻值。经作者慷慨授权转载图像 [64, 页。 4]。

CO_2 二氧化碳当量排放。请注意，应用基础模型的成本要低得多。OpenAI 对使用 GPT-3 处理莎士比亚全集（共 90 万个词）收费 72 美元。Google 和微软在大规模应用中使用基础模型，例如用于翻译或网络搜索。更详细的讨论见[13,第 139] 页。

基础模型可能导致失业和社会不平等

另一方面，基础模型在语言处理方面取得的突破性能力，可能使目前由受雇人类劳动者完成的任务实现自动化，例如回复客户服务咨询、翻译文档、编写计算机代码或创建图像，从而对就业产生负面影响。这意味着现有劳动者需要接受再培训，以胜任新的工作，并最终可能导致失业率上升。由于目前尚不清楚需要多少新的人类劳动者，经济风险难以预测。一个令人担忧的趋势是，智力要求较高的工作首次开始大规模地被机器取代 [5]。根据这项研究，风险最高的就业领域包括物流、办公室职员、生产、服务、销售和建筑业。

Paolillo等人 [64] 从一项观察出发：工作需要多种能力的组合。他们将职业能力分解为87种不同的技能，并估算这些技能的自动化风险 （自动化风险指数，ARI）。据此，他们计算出近1,000种职业的自动化

风险。自动化风险指数（ARI）可理解为某项工作所需的人工技能中，也能由机器完成的比例。作者估计，物理学家的自动化风险指数最低，为0.44；屠宰工和肉类包装工的自动化风险指数最高，为0.78。图8.4展示了不同职业群组的自动化风险指数估值。自动化风险指数的中位数约为0.6，这意味着所有技能中约有60%可以实现自动化。因此，自动化很可能会对几乎所有职业造成重大影响。作者认为，适度的职业再培训可以大幅降低劳动者面临的自动化风险。

人工智能与以往的创新不同，它不会自动化体力工作，而是自动化认知任务 Bordot [15]。作者使用涵盖33个经合组织国家的面板数据，研究了人工智能、机器人与失业之间的关联。他们发现，机器人和人工智能都倾向于推高失业率，为技术性失业相关文献提供了更多证据。他们还得出结论：在三年期间，人工智能会提高受教育程度中等者的失业率，而对其他人的影响则为负面或不显著。这表明，随着人工智能使用日益普及，中等技能工作受到的冲击最大。

基础模型非常擅长创作故事；合理推测，再过几年，它们就能以半自动的方式撰写整部小说或创作歌曲。同样，基础模型能够创作和修改图形及照片级真实图像，从而使平面设计师和摄影师的工作贬值。这对于创意作品（如虚构作品、新闻文章和音乐）尤为如此，对科学研究也同样如此。De- houche [30] 讨论了这类剽窃行为。由于无法以生成的内容侵犯版权为由提出异议，这一发展可能会损害创意或创新工作的盈利能力。虽然这种版权侵蚀可能造成危害，但也可能带来显著的社会收益，例如让更广泛的群体更容易获取教育或创意材料。版权突破所带来的潜在危害和收益值得进一步考量 [91]。与此同时，法院正在处理这一问题 [89]。

截至2021年1月，全球有46亿活跃互联网用户，占全球人口的59.5% [81]。然而，许多社会群体和国家将无法获取需要特殊高性能计算环境的基础模型。无法获得这项技术，可能会使某些群体获得不成比例的利益，从而延续全球不平等。对于不识字、未接受过完整教育或有学习困难的人来说，基础模型的应用——例如翻译、文本转语音和数字助理——尤其重要。在选择用于训练基础模型的语言时，应当考虑这一点。Bender等人 [7] 详细讨论了基础模型带来的利益与风险在全球范围内的分布。

基础模型可能促进统一的世界观和文化

目前，互联网由垄断企业主导。Alphabet掌控网络搜索，亚马逊主导电子商务，苹果是商务智能手机领域的领导者，Meta掌控社交网络，微软则控制商务软件 [6]。这些公司受益于巨大的规模经济，因为数字平台往往需要高昂的前期成本，但在完成初始投入后，为新增客户提供服务的成本几乎为零。此外，这些公司还通过收购初创企业和竞争对手来消除竞争者。

因此，可以推断，基础模型服务将融入这些垄断企业现有的基础设施，并利用现有搜索引擎作为信息提供组件。因此，世界上大多数文档的撰写很可能都将由少数几种不同的基础模型提供支撑。这意味着，少数原始模型的优势、创造力、偏见、缺陷、怪癖和特性将无处不在，并可能以一致的方式影响多种语言的文化 [13, p. 151]。这种均质化可以为大量应用带来极高的收益，但也可能对其他领域产生深远的负面影响。Kleinberg等人 [48] 将其称为“算法单一文化”，这可能导致偏见趋同、特定观点和理论得到推广、拒绝行为既一致又武断、分类错误，或个人及群体受到不公正对待。Cave等人 [22]甚至指出，无论是在日常新闻报道还是奇幻文学中，人工智能都主要被描绘成白色，因为白色显然仍与理性、智能和力量相关联。由于现行反垄断法对互联网公司不起作用，因此需要出台新的法规来拆分垄断企业 [6]。这就需要重新界定市场、提出服务互操作性要求，并将数据所有权转交给客户，使客户能够将数据转移给其他服务提供商。

基础模型的法律监管是必要的

基于在规模极其庞大的文本集合上训练的基础模型进行自动化应用，给我们的社会带来了一系列全新的挑战。我们希望拥有符合我们价值观、以人为本、运行可靠且同时具备竞争力的公共利益系统。因此，我们必须努力实现公平、客观的结果，并避免不良后果。应用对所有利益相关者的公平行为、对用户需求的考量、可靠、可理解且安全的运行，以及对敏感数据的保护，都是基础模型可信使用的核心要求。

众所周知，组织在采用新科技时往往会做出糟糕的决策 [19]。另一方面，Google 等商业公司没有直接的激励去提高透明度或减少社会不平等[72]。为了让基础模型更加人性化且值得信赖，社会需要理解应当适用哪些护栏机制、原则和边界

、应如何开发基础模型应用、应允许它们以多大程度自主行动，以及我们希望如何对其进行控制。因此，不同国家正在努力为基础模型和人工智能系统制定规则。

欧盟提出了一项基于人工智能应用 [34]相关风险的监管框架。该框架将风险划分为四个等级：风险极低或无风险、有限风险、高风险和不可接受风险。所有具有不可接受风险（威胁人们的安全、生计和权利）的人工智能系统都将被禁止 [13, p. 157]。高风险应用包括关键基础设施、教育培训、生物识别和安全组件，进入市场前必须通过若干严格的检查和评估。针对聊天机器人等有限风险系统，适用特别的透明度义务。人工智能电子游戏或垃圾邮件过滤器等风险极低或无风险的系统可以自由使用。目前欧盟使用的绝大多数人工智能系统都属于这一类别。

在美国，不同机构提出了具体的监管指南[83]。美国商务部正在制定“一套旨在实现可信人工智能系统的自愿性风险管理框架”。美国联邦贸易委员会列出了一系列合规要求，包括确保训练数据充足、测试模型以避免偏见、公开数据使用情况、如实呈现模型性能，以及提高建模目标的透明度。尽管目前尚无统一的人工智能监管规定，监管机构仍建议企业制定策略和流程，以实现设计即合规。这既能促进人工智能创新，也能确保系统的透明度和可解释性。此外，企业应定期审计和审查策略的使用情况，并记录这些流程，以遵守监管要求。Bommasani 等人对规范和监管进行了详细讨论 [13,p. 154]。

8.3 高级人工智能系统

自监督学习是基础模型中的标准方法，并在语言和图像识别任务中带来了前所未有的性能提升。然而，人类智能还有更多这一范式未涵盖的特征。本节首先讨论基础模型能否生成新的创意内容。接着，我们考察语言中的词语和概念如何“落地”，即与物理世界中相应的物体和过程建立联系。最后，我们探讨卡尼曼的人类行为理论，并讨论一些改进当前模型的思路。

8.3.1 基础模型能生成创新内容吗?

长期以来，人们一直在讨论一个问题：当前的基础模型能否生成创新的 内容，还是它们只是随机鹦鹉 [7] ，不假思索地重复从训练数据中获取的短语和文本片段。在《重启人工智能》一书中，Marcus 等人 [56] 也提出了类似的观点。他称 GPT-3 [43] “是一种令人惊叹的仿作生成方式，就像高中生抄袭时会改动个别词语，却并没有真正把各种观点融会贯通。它并不真正理解背后的观点。”如前所述，GPT-3 并不能真正“理解”它表达的内容，因为它无法将词语和短语与现实世界中的物体和事件联系起来。

Johnson 等人 [43] 使用句子“撰写一篇论文，讨论元小说在伊塔洛·卡尔维诺作品中的作用。”提示 GPT-3。该系统针对这一主题生成了一篇简洁的五段式总结。作者将生成的文本描述为“清晰且响应迅速”。当重复使用该提示时，GPT-3 会不断生成完全不同的新响应。当作者将每个生成的句子输入 Google 搜索引擎时，他一个也没有找到。每个句子都是针对该特定提示定制生成的。这说明基础模型非常擅长将不同的内容片段组合在一起。然而，它们并不是在字符串和单词层面运行，而是在上下文嵌入层面运行；上下文嵌入表达了短语和句子潜在的概念相似性，以及它们在大量句子和文档中的关系。

当我们考虑同时捕捉文本和图像内容的基础模型时，这一现象会更加明显。如第 7.2 节所述，DALL-E 2 等模型为图像块和文本词元构建了一个联合嵌入空间。在这一空间中，图像和文本并不是根据像素和字符串建立关联，而是根据这些图像块和词元的上下文相关嵌入建立关联。这些嵌入会根据图像和文本的整体构成而发生变化。生成新内容基于这些嵌入之间的相关性，因此能够创建图像与文本的新组合，例如对应于“一只柯基犬在演奏喷火小号”的图像（图 7.15），或者用于说明图像描述“时代广场上滑滑板的泰迪熊”的照片级真实图像（图 7.16）。尽管 DALL-E 2 不知道现实世界中的地点“Time Square”的任何物理属性，但它可以通过上下文嵌入的形式组合有关该地点的信息，并生成外观相当逼真的景象，而这些景象此前从未出现过。通过这种方式，基础模型确实能够生成创新内容。

8.3.2 在世界中实现语言接地

语言研究长期以来面临的一个问题是，机器如何才能“理解”语言的“意义”。Bender 等人 [8] 认为，“语言建模任务”仅将语言形式用作训练数据，因此原则上无法带来

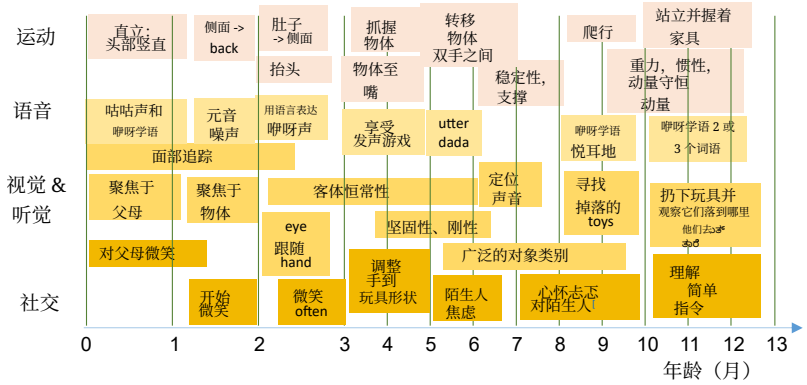


图 8.5 根据维基百科 [92] 和 LeCun [51]，婴儿感知能力发展的时间线。抽象的自然定律，例如物体会受到重力和惯性的影响，是在较简单的概念（如客体恒常性，以及将物体归入广泛类别）之后习得的。大多数知识是通过观察获得的，直接操作则很少，尤其是在最初几个月。

对意义的学习”。这里的“意义”被定义为语言形式与现实世界中的交际意图之间的关系。在此上下文中，语言建模是一个字符串预测系统。按照这一观点，当前的语言模型并未获得“意义”，而只是将一些短语与其他短语联系起来。

婴儿的感知学习也以自监督的方式进行（图 8.5）。在语言发展过程中，父母和婴儿会指向物体 [26], ，而婴儿在学习语言的许多其他方面之前，就会先学会与常见物体相关的词语的具身意义 [10]。婴儿只是观察周围环境，并可能会形成某种预期，认为环境（例如物体运动、视角变化）会随时间如何变化（图 8.6）。多次看到苹果落下，就足以让婴儿了解重力如何起作用。此外，物体不会因为离开视野而消失。婴儿可能会通过预测这些变化来学习，并在出现偏差时无意识地修正自己的预期[51]。这相当于在视频领域通过预测后续帧进行无监督学习。NÜWA 系统（第 7.3.4 节）已经以这种方式完成预训练，并在视频后续帧预测方面达到了最先进水平。

如果一个系统仅从词语层面进行训练，就很难学会一个概念。例如，仅仅知道狗与狗绳、耳朵、猫、哺乳动物、腿、毛皮、尾巴、玩具、吠叫等相关，并不能完全理解狗 [50]。信息必须经过结构化处理，使人们知道玩具是狗玩耍的东西，毛皮是狗的身体覆盖物，哺乳动物是狗所属的类别，诸如此类。狗的头部靠近四条腿，并不构成一只狗。因此，当狗以多种媒体形式出现时，最容易学会“狗”这一概念，例如以图像、词语或电影中的形象出现，在电影里它追逐一只猫。

最近，有人提出了一种名为 **PLATO** [67]的模型，旨在从视频中学习直觉物理。PLATO 将每个分割后的视频帧分解为一组物体



图 8.6 婴儿观察周围环境并操纵物体。它会形成一些预期，认为环境（例如物体运动、视角变化）会如何随时间演变。它会预测这些变化，并在出现偏差时不自觉地进行学习。图像来源见表 A.4。

并使用感知模块为每个物体分配一个 ID，以便随时间跟踪物体。借助预期违背标准，PLATO 可以学习多种物理概念，例如物体连续性、方向惯性、物体持存性和物体实体性。该模型的方法为将直觉物理概念建立在视觉感知之上提供了一种途径。

可以预期，自监督学习将通过纳入更多维度而得到扩展，例如三维、自主运动以及对环境的主动操控。正如 LeCun 所说：“不过，下一代人工智能不会学习语言或图像，而是直接从视频中学习。Meta 目前正投入大量精力，从第一人称视角收集视频数据，用于新一代人工智能[41], ，但 YouTube 视频也适合作为训练材料” [73]。LeCun 认为，人工智能系统能够从此类视频中学习我们世界的物理基础。进而，这种理解将成为众多能力的基础，例如抓取物体或驾驶汽车。

Bisk 等人 [12] 提供了更为详细的视角。作者认为，语言学习必须与“语言外事件”建立联系。他们区分了语言学习的不同词汇范围（图 8.7）。范围最窄的包括经过精心创建的语料库，例如人工标注的宾夕法尼亚树库。BERT 就是在这类经过精心整理的数据集上训练的。下一个范围涵盖网络规模数据集；以 PaLM 为例，其中包含 7800 亿个仅用于训练一次的词元。根据缩放定律（第 3.5.1 节），可以预期，随着数据量和模型参数的增加，语言预测原本就很高的准确率还会进一步提升。

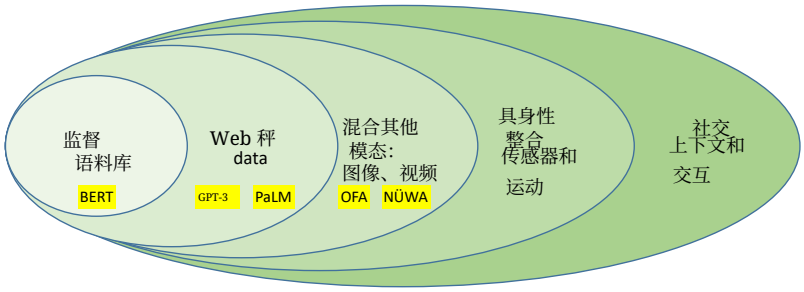


图 8.7 语言扎根的世界范围。虽然前三个范围已经在一定程度上得到探索，但剩下的两个范围仍需在未来加以考虑 [12]。

下一个范围是将语言与来自其他模态的感官输入相结合。例如，学习一幅画的意义、视觉印象及其寓意，就需要这种结合。利用将图像与图像描述相对应的数据集，是朝这一方向取得进展的好方法。当视频内容配有字幕，且还有语音或转录语音可用时，视觉印象、音频、语音和语言之间便能建立更多联系。OFA 和 NÜWA 模型就是这一范围的典型例子，但它们还有许多改进空间。

如果需要回答以下问题：“橙子更像棒球，还是更像香蕉？”，那么仅凭视觉外观是不够的。此时必须确定橙子的不同特征，例如重量、可移动性、可塑性、可变形性和味道。只有用手操控并探索橙子，才能做到这一点。此时需要下一个范围，在这一范围内，施事在世界中移动并采取动作，同时接收关于自主运动、力和身体位置的各种触觉与感官印象。只有通过这种方式，才能从交互中学习世界的基本物理属性。要在这一领域取得进展，需要基础模型与机器人学的收敛，PLATO 已开始推动这一进程。Thomason 等人 [84] 提出使用三维物体使语言落地。目前的方法仍相当有限。

最终的研究范围是人际沟通，这是自然语言的核心用途。目前尚不清楚计算机系统如何才能在社会情境中作为具身参与者发挥作用。小冰和 LaMDA 等对话模型是初步尝试。Bisk 等人 [12] 对这些问题进行了深入探讨，而这些问题可能在遥远的未来更具现实意义。

8.3.3 快思考与慢思考

智能思维以不同的速度运作。诺贝尔经济学奖得主丹尼尔·卡尼曼通过长期研究人类行为，提出了关于两种不同思维系统的假设 [45] （图 8.8）。系统 *1*（快思考）快速、凭直觉且带有情绪。示例包括理解一句简单的口语

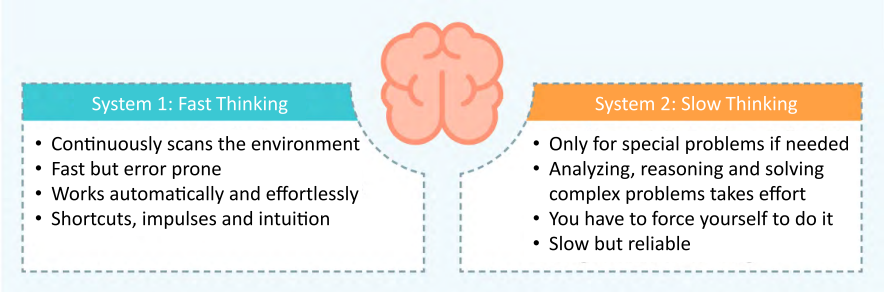


图 8.8 根据卡尼曼 [45]，人脑中用于快思考与慢思考的两个系统的性质。

句子、在安静的道路上驾驶汽车，或识别图片中的一个物体。系统1持续运作，并根据我们即时的感知产生印象、直觉和快速判断。

系统2（慢思考）速度较慢，更加审慎，也更具逻辑性。例如，它负责记住一个很久未见的人、将车停入狭窄的停车位，或解决算术问题16*34。只有当系统1出现问题，即无法很好地解释感知时，才会使用系统2。

与大脑中的系统2相对应的是一种容量有限的工作记忆 [32]。它可以在短时间内存储思维内容，并同时对其进行操作。它显然在问题求解和逻辑推理中发挥着重要作用。能够同时处理的信息单元数量估计在五到七个之间。人类能够意识到系统2的思维过程，而系统1的处理过程在很大程度上是潜意识的。系统2要求具备考虑世界抽象表征的能力。这涉及聚焦于有限的一组特征，并对其进行深度处理，同时忽略其他特征 [14]。

8.3.4 规划策略

图灵奖得主 Yann LeCun [53] 认为，当前的基础模型已经能够像系统1一样处理环境的许多方面。自监督学习能够很好地捕捉语音和语言，并将二者相互转换。图像也可以进行分析，并与文字描述相关联，但效果稍逊。视频、语音和文本的联合处理前景可期，但仍需进一步发展。

直到最近，基础模型才具备规划能力（第 7.4 节），即系统地面向未来，考虑目标、手段以及实现未来目标的方法。这对应于卡尼曼所说的系统2。基础模型基本上执行模型预测控制，并对系统进行模拟，模拟范围涵盖

一系列时间步 [74]。例如，沿着道路驾驶汽车。在此过程中，系统会同时模拟系统的状态（例如汽车的位置和速度）、动作（例如转动方向盘、加速）以及奖励（例如与球门的距离、与障碍物的距离）。基础模型使用一组观测轨迹进行训练，能够学习状态、动作与相应奖励之间的依赖关系。随后，它便能够预测下一步动作，以达到特定的奖励水平。使用基础模型进行规划时，已经可以纳入多模态，例如将图像用作状态描述来执行控制。

根据 Yann LeCun 的说法，“构建世界模型的能力基本上就是智能的本质” [53]。这些模型不仅需要预测物理运动，还需要预测人类行为、经济活动等。未来十年，人工智能面临的重大挑战是如何学习世界的预测模型，以应对不确定性。

LeCun 认为，这并不直接需要基于形式逻辑的推理，因为这种推理与高效学习所需的梯度不兼容。Yoshua Bengio 表示 [29], ：“有人认为，有些问题是神经网络无法解决的，我们必须求助于经典人工智能的符号方法。但我们的研究表明，事实并非如此。”推理更有可能是通过内部模拟和类比来完成的。正如 Geoffrey Hinton 所说：“但我猜最终我们会意识到，符号只存在于外部世界中，而我们在内部对大的向量进行运算。” [39]。值得注意的是，PaLM 等较新的模型使用思维链提示，其推理能力可以与普通人相当（第 4.2.3 节）。语言对于动物的智能也并不重要，它是在进化过程中较晚才获得的。

LeCun设想了一个复杂系统，其中某个高层“配置器”会针对当前问题实例化世界模型并执行心理模拟 [95]。他提出，存在一个世界模型引擎，可根据当前任务进行动态配置 [95]。这样一来，关于环境如何运作的知识便可以在不同任务间共享。一个关键要求是，世界模型必须能够表示并比较对环境的多种可能预测。这个配置器能够组合不同模型，并学习复杂的层级动作序列。在他的概念论文中，Yann LeCun [95]讨论了这种可能系统的许多细节。

结合语言、图像和控制的GATO模型或许是朝这个方向迈出的第一步，但仍处于起步阶段（第7.4.2节）。SayCan [2] 系统是一种将机器人与基础模型相结合的方法，能够用语言表达机器人的技能性质，例如“拾起海绵”。给定一项真实世界任务的描述，SayCan能够生成一系列技能执行步骤来完成任务。同样，许多强化学习领域的研究人员认为，最大化总奖励或许足以理解智能及其相关能力 [78]。

Melanie Mitchel 同意 LeCun 的观点，认为当前的基础模型还不够完善。“它们缺乏内存以及实际上非常重要的内部世界模型，”她说 [40]。原则上，这些模型并不需要语言。

但语言有一个很大的优势：只需加入一些事实或陈述，就能像少样本技术一样动态改变目标。总体而言，可以预期未来几年沿着这些发展方向将取得重大进展。

参考文献

[1] M. Abadi、A. Chu、I. Goodfellow、H. B. McMahan、I. Mironov、K. Talwar 和 L. Zhang。《差分隐私深度学习》。载于：2016 年 ACM SIGSAC 计算机与通信安全会议论文集。2016 年，第 308–318 页。[2] M. Ahn 等人。《照我所能而非照我所说：在机器人可供性中 grounding 语言》。2022 年 8 月 16 日。arXiv：2204.01691 [cs]。[3] F. Alam 等人。《多模态虚假信息检测综述》。2021 年。arXiv：2103. 12541。[4] W. An、Y. Guo、Y. Bian、H. Ma、J. Yang、C. Li 和 J. Huang。《MoDNA：面向 DNA 语言模型的基序导向预训练》。载于：第 13 届 ACM 生物信息学、计算生物学与健康信息学国际会议论文集。BCB '22。美国纽约州纽约：计算机协会，2022 年 8 月 7 日，第 1–5 页。ISBN：978-1-4503-9386-7。doi：10.1145/3535508.3545512。[5] W. Apt 和 K. Priesack。《人工智能与工作——机遇与风险并存》。载于：《人工智能：科技 | 应用 |社会》。V. Wittpahl 编。柏林、海德堡：施普林格，2019 年，第 221–238 页。ISBN：978-3-662-58042-4。doi：10.1007/978-3-662-58042-4_14。[6] Z. Arnao。《为什么垄断企业统治互联网，以及我们如何阻止它们》。2022 年 1 月 4 日。网址：
h
t
t
p://
u
chicagogate.com/articles/2022/1/4/why-monopolies-rule-internet-and-how-we-can-stop-them/（访问于 2022 年 4 月 26 日）。[7] E. M. Bender、T. Gebru 和 A. McMillan-Major。《随机鹦鹉的危险：语言模型可能太大了吗》。载于：FAccT 论文集（2021 年）。[8] E. M. Bender 和 A. Koller。《迈向 NLU：数据时代的意义、形式与理解》。载于：第 58 届计算语言学协会年会论文集。ACL 2020。在线：计算语言学协会，2020 年 7 月，第 5185–5198 页。doi：10.18653/v1/2020.acl-main.463。[9] Y. Bengio、Y. Lecun 和 G. Hinton。《面向人工智能的深度学习》。载于：《ACM通讯》64.7（2021 年），第 58–65 页。[10] E. Bergelson 和 D. Swingley。《在人类婴儿 6–9 个月大时，他们已知晓许多普通名词的意义》。载于：《美国国家科学院院刊》109.9（2012 年），第 3253–3258 页。[11] BigScience。《BigScience 大语言模型训练启动》。2021 年。网址：https: //bigscience.huggingface.co/blog/model-training-launched（访问于 2022 年 4 月 26 日）。[12] Y. Bisk 等人。《经验使语言扎根》。2020 年。arXiv：2004.10151。[13] R. Bommasani 等人。《基础模型的机遇与风险》。2021 年。arXiv：2108.07258。[14] G. Booch 等人。《人工智能中的快思考与慢思考》。载于：AAAI 人工智能会议论文集。第 35 卷。第 17 期。2021 年，第 15042–15046 页。[15] F. Bordot。《人工智能、机器人与失业：来自经合组织国家的证据》。载于：《创新经济学与管理学杂志》37.1（2022 年 1 月 21 日），第 117–138 页。ISSN：2032-5355。网址：https://www.cairn.info/revue-journal-of-innovation-economics-2022-1-page-117.htm（访问于 2022 年 4 月 25 日）。[16] T. F. Bresnahan 和 M. Trajtenberg。《通用目的技术"增长引擎"？》。载于：《经济学杂志》65.1（1995 年），第 83–108 页。[17] T. B. Brown 等人。《语言模型是少样本学习器》。2020 年。arXiv：2005.14165。

[18] B. Buchanan、A. Lohn、M. Musser 和 K. Sedova。《真相、谎言与自动化：语言模型如何改变虚假信息》。2021 年 5 月 1 日。网址：https://cset.georgetown.edu/publication/truth-lies-and-automation/（访问于 2021 年 10 月 13 日）。[19] R. Calo 和 D. K. Citron。《自动化行政国家：合法性危机》。载于：Emory LJ 70（2020 年），第 797 页。[20] N. Carlini。《污染无标签数据集的 ${$半监督$}$ 学习》。载于：第 30 届 USENIX 安全研讨会。USENIX 安全 21。2021 年，第 1577–1592 页。[21] N. Carlini 等人。《从大型语言模型中提取训练数据》。2021 年 6 月 15 日。arXiv：2012.07805。[22] S. Cave 和 K. Dihal。《人工智能的白性》。载于：《哲学与技术》33.4（2020 年 12 月 1 日），第 685–703 页。ISSN：2210-5441。doi：10.1007/s13347-020-00415-6。[23] S. Cen 和 D. Shah。《监管社交媒体上的算法过滤》。载于：神经信息处理系统进展 34（2021 年）。[24] A. Chowdhery 等人。《PaLM：借助 Pathways 扩展语言建模》。2022 年 4 月 5 日。arXiv：2204.02311 [cs]。[25] R. Chowdhury、N. Bouatta 和 S. Biswas。《使用语言模型和深度学习进行单序列蛋白质结构预测》。载于：《自然·生物技术》（2022 年 10 月 3 日），第 1–7 页。网址：https://www.nature.com/articles/s41587-022-01432-w（访问于 2022 年 10 月 14 日）。[26] C. Colonnesi、G. J. J. Stams、I. Koster 和 M. J. Noom。《指示行为与语言发展之间的关系：一项荟萃分析》。载于：《发展评论》30.4（2010 年），第 352–366 页。[27] A. D'Ulizia、M. C. Caschera、F. Ferri 和 P. Grifoni。《假新闻检测：评估数据集综述》。载于：《PeerJ 计算机科学》7（2021 年 6 月 18 日），e518。ISSN：2376-5992。doi：10.7717/peerj-cs.518。[28] E. Dayanik 和 S. Padó。《掩盖行为主体信息可带来更公平的政治主张检测》。载于：第 58 届计算语言学协会年会论文集。ACL 2020。在线：计算语言学协会，2020 年 7 月，第 4385–4391 页。doi：10.18653/v1/ 2020.acl-main.404。[29]《面向人工智能的深度学习》。与 Y. Bengio、Y. LeCun 和 G. Hinton 合作。2021 年 5 月 25 日。网址：https://vimeo.com/554817366（访问于 2022 年 4 月 27 日）。[30] N. Dehouche。《大规模生成式预训练变换器（GPT-3）时代的剽窃》。载于：《伦理学、科学、环境与政治》21（2021 年），第 17–23 页。[31] J. Devlin、M.-W. Chang、K. Lee 和 K. Toutanova。《BERT：面向语言理解的深度双向 Transformer 预训练》。2018 年。arXiv：1810.04805。[32] A. Diamond。《执行功能》。载于：《心理学年度评论》64（2013 年），第 135–168 页。[33]《经济学人》。《庞大的"基础模型"正在加速人工智能进步》。载于：《经济学人》（2022 年 6 月 11 日）。ISSN：0013-0613。网址：https://www.economist.com/ interactive/briefing/2022/06/11/huge-foundation-models-are-turbo-charging-ai-progress（访问于 2022 年 6 月 20 日）。[34] 欧盟。《人工智能监管框架 | 塑造欧洲的数字未来》。2021 年。网址：https: //digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai（访问于 2022 年 4 月 26 日）。[35] V. Feldman 和 C. Zhang。《神经网络记忆什么以及为何记忆：通过影响力估计发现长尾》。2020 年。arXiv：2008.03703。[36] A. Galen。TensorFlow Privacy。tensorflow，2021 年 11 月 12 日。网址：https://github.com/tensorflow/ privacy（访问于 2021 年 11 月 14 日）。[37] I. Garrido-Muñoz、A. Montejo-Ráez、F. Martínez-Santiago 和 L. A. Ureña-López。《深度自然语言处理中的偏见综述》。载于：《应用科学》11.7（2021 年），第 3184 页。[38] H. Gonen 和 Y. Goldberg。《给猪涂口红：去偏方法掩盖了词嵌入中的系统性性别偏见，却无法消除这些偏见》。2019 年 9 月 24 日。arXiv：1903.03862 [cs]。[39] K. Hao。《人工智能先驱 Geoff Hinton："深度学习将能够做到一切"》。《麻省理工科技评论》。2020 年 11 月 3 日。网址：https://www.technologyreview.com/2020/11/03/1011616/ai-godfather-geoffrey-hinton-deep-learning-will-do-everything/（访问于 2022 年 3 月 28 日）。[40] M. Heikkilä 和 W. D. Heaven。《Yann LeCun 为人工智能的未来提出大胆的新愿景》。《麻省理工科技评论》。2022 年 6 月 24 日。网址：https://www.technologyreview.com/2022/06/24/1054817/yann-lecun-bold-new-vision-future-ai-deep-learning-meta/（访问于 2022 年 7 月 10 日）。

[41] C. Jawahar. 教会人工智能通过你的眼睛感知世界。2021年10月14日。网址：https : / / ai . facebook . com/ blog / teaching - ai - to - perceive - the - world - through - your- eyes/（访问于 10/25/2021）。[42] Y. Ji、Z. Zhou、H. Liu 和 R. V. Davuluri。“DNABERT：用于基因组 DNA语言的 Transformer 模型预训练双向编码器表示”。载于：Bioinformatics 37.15（2021年），第 2112–2120 页。[43] S. Johnson 和 N. Iziev。“人工智能正在掌握语言。我们应该相信它所说的话吗？”载于：《纽约时报》杂志（2022年4月15日）。ISSN：0362-4331。网址：https://www. nytimes.com/2022/04/15/magazine/ai-language.html（访问于 04/26/2022）。[44] J. Jumper 等人。“借助 AlphaFold 实现高精度蛋白质结构预测”。载于：《自然》596.7873（7873年8月，2021年），第 583–589 页。ISSN：1476-4687。DOI：10.1038/s41586-021-03819-2。[45] D. Kahneman。《思考，快与慢》。Macmillan，2011年。[46] D. Kahneman 和 A. Tversky。“论预测心理学”。载于：Psychol. Rev. 80.4（1973年），第 237 页。[47] T. Khan、A. Michalas 和 A. Akhunzada。“2021年假新闻暴发：我们能阻止病毒式传播吗？”载于：《Journal of Network and Computer Applications》190（2021年9月15日），第 103112 页。ISSN：1084-8045。DOI：10.1016/j.jnca.2021.103112。[48] J. Kleinberg 和 M. Raghavan。“算法单一文化与社会福利”。载于：《美国国家科学院院刊》118.22（2021年）。[49] S. Kumar、S. Kumar、P. Yadav 和 M. Bagri。“假新闻检测技术分析综述”。载于：2021年国际人工智能与智能系统会议 ICAIS。2021年人工智能与智能系统国际会议（ICAIS）。2021年3月，第 894–899 页。DOI：10.1109/ICAIS50930.2021.9395978。[50] B. M. Lake 和 G. L. Murphy。“心智与机器中的词义”。载于：Psychol. Rev.（2021年）。[51] Y. LeCun。“基于能量的自监督学习”。载于：（2019年11月19日），第 68 页。网址：http://helper.ipam.ucla.edu/publications/mlpws4/mlpws4_15927.pdf。[52] D. Lewis、A. Zugarini 和 E. Alonso。“用于英文诗歌生成的音节神经语言模型”。载于：Conf. Comput. Creat.（2021年），第 7 页。[53] Lex Fridman，导演。Yann LeCun：智能暗物质与自监督学习 | Lex Fridman 播客 #258。2022年1月22日。网址：https://www.youtube.com/watch?v=SGzMElJ11Cc（访问于 04/26/2022）。[54] R. Lim、M. Wu 和 L. Miller。为你的应用定制 GPT-3 14, 2021 3 02 16 2022。OpenAI。Dec. . 网址：https://openai.com/blog/customized-gpt- /（访问于 / / ）。[55] Z. Lin 等人。“进化尺度上的蛋白质序列语言模型可实现精准结构预测”。载于：bioRxiv（2022年）。[56] G. Marcus 和 E. Davis。《重启人工智能：构建我们能够信任的人工智能》。Vintage，2019年。[57] N. Mehrabi、F. Morstatter、N. Saxena、K. Lerman 和 A. Galstyan。“机器学习中的偏见与公平性综述”。载于：ACM Comput. Surv. CSUR 54.6（2021年），第 1–35 页。[58] D. Meyer。伪造的基辅领导人克利奇科视频骗过了欧洲各地的市长——但目前尚不清楚这是否真的是一段“深度伪造”视频。《财富》。2022年6月27日。网址：https://fortune.com/
2
0
2
2/
06/27/fake-kyiv-klitschko-giffey-ludwig-martinez-almeida-karacsony-colau-deepfake-ai/（访问于 07/09/2022）。[59] F. Mireshghallah、M. Taram、P. Vepakomma、A. Singh、R. Raskar 和 H. Esmaeilzadeh。“深度学习中的隐私：综述”。2020年。arXiv：2004.12254。[60] S. Mo 等人。“针对不同细胞类型的调控基因组多模态自监督预训练”。2021年。arXiv：2110.05231。[61] W. Nagel。欧洲人工智能语言模型项目 Open GPT-X 启动。德累斯顿工业大学。2022年1月20日。网址：https : / / tu - dresden . de / tu - dresden / newsportal / news / projektstart - open - gpt - x?set_language=en（访问于 04/21/2022）。[62] A. Nichol 等人。“GLIDE：通过文本引导的扩散模型实现照片级图像生成与编辑”。2021年。arXiv：2112.10741。

[63] D. Ofer、N. Brandes 和 M. Linial。“蛋白质的语言：自然语言处理、机器学习与蛋白质序列”。载于：Comput. Struct. Biotechnol. J. 19（2021年），第 1750–1758 页。[64] A. Paolillo 等人。“通过评估工作自动化风险和韧性替代方案，如何与机器人竞争”。载于：Sci. Robot. 7.65（2022年4月13日），eabg5561。DOI：10.1126/ scirobotics.abg5561。[65] D. Paperno 等人。“LAMBADA 数据集：需要广泛话语上下文的词语预测”。2016年6月20日。arXiv：1606.06031 [cs]。[66] I. Perov 等人。“DeepFaceLab：集成、灵活且可扩展的换脸框架”。2021年6月29日。arXiv：2005.05535 [cs, eess]。[67] L. S. Piloto、A. Weinstein、P. Battaglia 和 M. Botvinick。“受发展心理学启发的深度学习模型中的直觉物理学习”。载于：Nat Hum Behav（2022年7月11日），第 1–11 页。ISSN：2397-3374。DOI：10.1038/s41562-022-01394-8。[68] C. Qu、W. Kong、L. Yang、M. Zhang、M. Bendersky 和 M. Najork。“借助隐私保护 BERT 实现自然语言理解”。载于：第 30 届 ACM 国际信息与知识管理会议论文集。2021年，第 1488–1497 页。[69] A. Radford、J. Wu、R. Child、D. Luan、D. Amodei 和 I. Sutskever。“语言模型是无监督多任务学习者”。载于：OpenAI博客 1.8（2019年），第 9 页。[70] J. W. Rae 等人。“扩展语言模型：Gopher 训练方法、分析与见解”。载于：ArXiv 预印本 ArXiv211211446（2021年12月8日），第 118 页。[71] Rasa。为什么选择 Rasa？Rasa。2022年。网址：https://rasa.com/product/why-rasa/（访问于 04/21/2022）。[72] R. Reich 和 J. Weinstein。《系统错误：大型科技公司错在哪里，以及我们如何重启 | 政治学》。HarperCollins，2021年。网址：https://politicalscience.stanford. edu/ publications/ system- error- where- big- tech- went- wrong- and- how- we- can- reboot（访问于 04/26/2022）。[73] M. Schreiner。Meta 人工智能负责人：人工智能面临的三大挑战。MIXED。2022年1月29日。网址：https://mixed-news.com/en/metas-ai-chief-three-major-challenges-of-artificial-intelligence/（访问于 02/06/2022）。[74] M. Schwenzer、M. Ay、T. Bergs 和 D. Abel。“模型预测控制综述：工程视角”。载于：Int J Adv Manuf Technol 117.5-6（2021年11月），第 1327–1349 页。ISSN：0268-3768, 1433-3015。DOI：10.1007/s00170-021-07682-3。[75] J. Sevilla、L. Heim、A. Ho、T. Besiroglu、M. Hobbhahn 和 P. Villalobos。机器学习三个时代的计算趋势。2022年3月9日。DOI：10.48550/arXiv.2202.05924。arXiv：2202.05924 [cs]。[76] S. M. Shifath、M. F. Khan 和 M. Islam。“一种打击新冠肺炎假新闻的 Transformer 方法”。2021年。arXiv：2101.12027。[77] K. Shuster、S. Poff、M. Chen、D. Kiela 和 J. Weston。“检索增强可减少对话中的幻觉”。2021年。arXiv：2104.07567。[78] D. Silver、S. Singh、D. Precup 和 R. S. Sutton。“奖励即足够”。载于：人工智能 299（2021年10月1日），第 103535 页。ISSN：0004-3702。DOI：10.1016/j.artint.2021.103535。[79] I. Solaiman 和 C. Dennison。“借助价值观导向数据集使语言模型适应社会的过程（Palms）”。载于：神经信息处理系统进展 34（2021年）。[80] K. Stahl。“社交媒体中的假新闻检测”。载于：（2018年5月15日），第 6 页。[81] statista。2021年全球互联网用户。Statista。2021年。网址：https://www.statista.com/statistics/ 617136/digital-population-worldwide/（访问于 04/25/2022）。[82] H. Sun 等人。“对话模型安全性：分类体系、数据集与基准”。2021年。arXiv：2110.08466。[83] H. Sussman、R. McKenney 和 A. Wolfington。美国人工智能监管逐渐成形。2021年11月18日。网址：https:// www.orrick.com/ en/ Insights/ 2021/ 11/ US-Artificial- Intelligence-Regulation-Takes-Shape（访问于 04/26/2022）。[84] J. Thomason、M. Shridhar、Y. Bisk、C. Paxton 和 L. Zettlemoyer。“借助三维物体实现语言落地”。载于：（2021年），第 11 页。[85] R. Thoppilan 等人。“LaMDA：用于对话应用的语言模型”。2022年2月10日。arXiv：2201.08239 [cs]。

[86] G. Todorov。《2021年及以后人工智能的65项统计数据》。Semrush 博客。2021年2月26日。网址：https : / / www . semrush . com / blog / artificial - intelligence - stats（访问于2022年3月28日）。[87] R. Toews。《价值十亿美元的语言人工智能初创企业浪潮即将到来》。《福布斯》。2022年3月27日 2022 03 27。网址：https://www.forbes.com/sites/robtoews/ / / / a-wave-of-billion-dollar-language-ai- startups-is-coming/（访问于2022年4月20日）。[88] A. Vaswani 等人。《Attention Is All You Need》。载于：神经信息处理系统进展，2017年，第5998–6008页。[89] S. J. Vaughan-Nichols。《GitHub 的 Copilot 面临首起开源版权诉讼》。2022年11月11日。网址：https://www.theregister.com/2022/11/11/githubs_copilot_opinion/（访问于2022年12月17日）。[90] K. Wali。《EleutherAI 发布 GPT-NeoX-20B：最大的公共访问语言模型》。《Analytics India Magazine》。2022年2月14日。网址：https://analyticsindiamag.com/eleutherai-launches-gpt-neox-20b-the-biggest-public-access-language-model/（访问于2022年2月23日）。[91] L. Weidinger 等人。《语言模型造成伤害的伦理与社会风险》。12月 8, 2021 2112 04359 [ ] 。arXiv：。cs。

[92] 维基百科。《儿童发展阶段》。载于：维基百科。2023年1月15日。网址：https://en.wikipedia. org/w/index.php?title=Child_development_stages&oldid=1133768924（访问于2023年1月22日）。[93] Woebot。Woebot Health。Woebot Health。2022年。网址：https://woebothealth.com/（访问于2022年4月21日）。[94] M. Xiao 和 P. Mozur。《数字追捕：中国警方如何在 Twitter 和 Facebook 上追踪批评者》。载于：《纽约时报》。科技（ 2021 年12月31日）。ISSN：0362-4331。网址：https://www.nytimes.com/2021/12/31/technology/china- internet- police- twitter.html（访问于2022年4月25日）。[95] Yann LeCun，导演。Yann LeCun：《迈向自主人工智能的道路》，百度，2022-02-22。2022年2月25日。网址：https://www.youtube.com/watch?v=DokLw1tILlw（访问于2022年4月26日）。[96] A. Yousefpour 等人。《Opacus：PyTorch 中易于使用的差分隐私库》。2021年。arXiv：2109.12298。[97] P. Yu、Z. Xia、J. Fei 和 Y. Lu。《深度伪造视频检测综述》。载于：IET Biom. 10.6（2021年），第607–624页。[98] R. Zellers、A. Holtzman、H. Rashkin、Y. Bisk、A. Farhadi、F. Roesner 和 Y. Choi。《抵御神经网络假新闻》。2020年12月11日。arXiv：1905.12616。[99] R. Zhang 等人。《Youling：人工智能辅助歌词创作系统》。2022年。arXiv：2201.06724。[100] S. Zhang 等人。《OPT：开放式预训练Transformer语言模型》。2022年5月5日。arXiv：2205.01068 [cs]。[101] J. Zhao、Y. Zhou、Z. Li、W. Wang 和 K.-W. Chang。《学习性别中立词嵌入》。2018年8月29日。arXiv：1809.01496 [cs, stat]。[102] L. Zhou、J. Gao、D. Li 和 H.-Y. Shum。《小冰的设计与实现：共情型社交聊天机器人》。载于：计算语言学。46.1（2020年），第53–93页。[103] X. Zhou、M. Sap、S. Swayamdipta、Y. Choi 和 N. Smith。《有害语言检测自动去偏的挑战》。载于：计算语言学协会欧洲分会第16届会议论文集主卷。计算语言学协会欧洲分会第16届会议论文集：主卷。线上：计算语言学协会，2021年，第3143–3155页。doi：10.18653/v1/2021.eacl-main.274。

附录 A 附录

A.1 图形中所用图像的来源与版权

表 A.1 第 1—3 章中使用的图像来源。

图	引用图像
图 2.18 and 图 8.1	文本：https://de.freepik.com/vektoren-kostenlos/schulbuecher-elemente-set_9387094.htm“学校书籍-书籍元素套装”由 macrovector 创作的矢量图形 - de.freepik.com，已裁剪 图像：https://de.freepik.com/免费矢量图/视觉愿景板卡通片插画与旅行和-家庭符号_13916392.htm“愿景板卡通插画，包含旅行和家庭象-bolen”由 macrovector 创作的矢量图形 - de.freepik.com，已裁剪 语音：https://de.freepik.com/vektoren-kostenlos/eine-junge-junge-singt-mit-mikrofon_20708335.htm“一名年轻女子用麦克风唱歌”由 brgfx 创作的矢量图形 - de.freepik.com，已裁剪 结构化数据：https://de.freepik.com/vektoren-kostenlos/tabellenkalkulation-in-笔记本电脑-und-台式电脑图标_24800197.htm“笔记本电脑和台式电脑图标中的电子表格”向量图形由 gstudioimagen1 创建 - de.freepik.com，已裁剪 三维形状：https://de.freepik.com/vektoren-kostenlos/geometrische-3d-formen-halbkugel-oktaeder-球体和环面、圆锥体、圆柱体与棱锥_10412342.htm“几何三维形状：半球、八面体、球体和环面、圆锥体、圆柱体与棱锥”矢量图形由 upklyak 创建 - de.freepik.com，已裁剪 对话：https://de.freepik.com/vektoren-kostenlos/freunde-treffen-sich-zum-hobbyspielkartenspiel_24077825.htm“朋友相聚玩爱好纸牌游戏”矢量图形由 upklyak 创作 - de.freepik.com，已裁剪 视频：https://de.freepik.com/vektoren-kostenlos/buendel-gesetzte-ikonen-der-kinounterhaltung_5720507.htm 由 gstudioim-创作的“电影娱乐图标套装”矢量图形 agen - de.freepik.com，已裁剪 控制：https://de.freepik.com/vektoren-kostenlos/isometrischen-strasse-mit-einem-roten-auto 965677.htm_“带红色汽车的等距街道”由以下人员创作的矢量图形 Freepik - de.freepik.com，已裁剪 训练：https://de.freepik.com/vektoren-kostenlos/leute-die-abenteueraktionen-machen_3065118.htm“进行冒险活动的人”矢量图形由 pikisuperstar 创作 - de.freepik.com，已裁剪 基金会模型：https://de.freepik.com/vektoren-kostenlos/globales-网络-连接区域-社交媒体-全球概念_4611150.htm“全球 networking-verbindungsbereich-社交媒体-weltweites konzept”矢量图形由宏向量-tor_官方 - de.freepik.com，已裁剪 搜索引擎：https://de.freepik.com/vektoren-kostenlos/netzwerk-datenbank-konzept_1531128.htm 由 macrovector 创作的“网络数据库概念”矢量图形 - de.freepik.com，已裁剪 问题回答：https://de.freepik.com/vektoren-kostenlos/verschiedene-leute-die-fragen-各类提问者插图_13244082.htm“各种提问者插图”矢量图形由 freepik 创作 - de.freepik.com 情感：https://www.flaticon.com/de/kostenloses-icon/gluecklich_187130“表情符号图标”由 Roundicons - Flaticon 创作，已裁剪 https://www.flaticon.com/de/kostenloses-icon/traurig_187143“悲伤图标”由 Pixel perfect - Flaticon 创作 信息抽取：https://de.freepik.com/vektoren-kostenlos/tatort-zusammensetzung_6168610.htm 由 macrovector 创作的“犯罪现场构图”矢量图形 - de.freepik.com，裁剪 图像图像描述：https://de.freepik.com/vektoren-kostenlos/kreatives-stimmungsbrett-in-柔和色调_6155033.htm“柔和色调的创意情绪板”矢量图形，创作者：coolvector - de.freepik.com，已裁剪 物体识别：https://de.freepik.com/vektoren-kostenlos/unterschiedliches-haustierkonzept_7970801.htm“不同的宠物概念”由 pikisuperstar 创作的矢量图形 - de.freepik.com，已裁剪 说明如下：https://de.freepik.com/vektoren-kostenlos/digitale-uhr-mit-streetmap-auf-屏幕 814679.htm_“屏幕上带有街道地图的数字时钟”矢量图形由 rocketpixel 创作 - de.freepik.com，已裁剪 图像生成：https://de.freepik.com/vektoren-kostenlos/kuenstler-malt-seine-思想-auf-画布_8354900.htm“艺术家将思想画在画布上”矢量图形创作自 pikisuperstar - de.freepik.com，已裁剪 视频创作：https://de.freepik.com/vektoren-kostenlos/电影构图-mit-演员-in- posisjon 在太空背景导演上为技术人员制作服装矢量插图_4359258.htm“由穿着服装的演员在太空背景导演的电影构图，搭配技术人员”矢量图形由 macrovector 创作 - de.freepik.com，已裁剪
图 2.19	“二维梯度下降”，作者：Gpeyre https://commons.wikimedia.org/wiki/File:Gradient_Descent_in_2D.webm 知识共享署名-相同方式共享 4.0
图 3.23	发动机：公共领域https://openclipart.org/detail/295364/4stroke-engine-cycle 鹰：公共领域 https://openclipart.org/detail/252471/soaring-eagle-no-background

表 A.2 第 6 章中所用图像的来源。

图	引用图像
图 6.5	男人：https://www.flaticon.com/free-icon/man_702023 “男人图标”由 monkik 创建 - Flaticon 新闻：https://de.freepik.com/freie-ikonen/zeitung_14362542.htm “新闻图标”由 Prosymbols 创作 - Flaticon 文档：https://www.flaticon.com/free-icon/documents_1181771 “文档图标”由 Freepik 创作 - Flaticon 维基百科：https://de.wikipedia.org/wiki/Datei:Wikipedia-logo.png 维基百科 徽标，方形，no 文本版本 1，由 Nohat（概念由 Paullusmagnus 提出）https://creativecommons.org/licenses/by-sa/3.0/deed.de
图 6.11	截取自 动画 GIF 中的 https://ai.googleblog.com/2020年/06/RECENT-advances-in-google-translate.html
图 6.20	~~数据库：https://de.freepik.com/vektoren-kostenlos/netzwerkserver-eingestellt_3924742.htm “网络服务器套装”~~ 网络服务器套装”矢量图形由 macrovector 创作，来源于 de.freepik.com，已裁剪 人物：图像由 studiogstock 上传至 Freepik。 https://de.freepik.com/vektoren-kostenlos/gruppe-von-带对话气泡的人物_5825572.htm 扩音器：可根据 Pixabay 许可免费使用。 https://pixabay.com/vectors/icon-loudspeaker-speaker-horn-1628258/ 麦克风：根据 Pixabay 许可免费使用。https://pixabay.com/vectors/mic-microphone-record-sound-audio-1296056/
图6.21	云： https://de.freepik.com/vektoren-kostenlos/satz-von-wolken-des-vektors-3d_17962159.htm “3D矢量云集合”矢量图形，由 vectorom 创建 - de.freepik.com，已裁剪
图6.23	女性： https://de.freepik.com/vektoren-kostenlos/abstrakte-hand-gezeichnete-frauenportraetsammlung_12978840.htm “抽象手绘 绘制的女性肖像集”由 Freepik 创作的矢量图形 - de.freepik.com，已裁剪

表 A.3 第7章中所用图像的来源。

图	引用图像
图 7.1	梅尔频率倒谱系数：使用 Python 脚本自行生成的图，见 https://haythamfayek.com/2016/04/21/speech-用于机器学习的处理.html
图 7.2	MFCC：使用 Python 脚本自行生成的图，来源：https://haythamfayek.com/2016/04/21/speech-用于机器学习的处理.html
图 7.3	Chris Sloan 拍摄的“阿代尔的孩子与乌鸫”。已裁剪并添加目标框。https://www.flickr.com/photos/sloanpix/14745258296/ 采用（知识共享署名 2.0）许可
图 7.4	Helder da Rocha 创作的“科隆大教堂”。已分割。https://www.flickr.com/photos/helder/167319167/in/照片列表 采用（CC BY-SA 2.0）许可
图 7.9	棒球：“2017年9月18日波士顿红袜队对巴尔的摩金莺队”，作者为来自美国马里兰州汉诺威的 Keith Allison——Mookie Betts。知识共享署名-相同方式共享 2.0。https://en.wikipedia.org/wiki/Baseball#/media/File:Mookie_Betts_击打_the_球_(36478781664).jpg 公共汽车：“蒙哥马利县校车 [02]“，作者为 Ben Schumin（CC BY-SA 2.0）https://www.flickr.com/photos/schuminweb/12275445544/ 手推车：“干草车上的农民 - 马拉谷 - 马拉穆列什 - 罗马尼亚”，作者为 Adam Jones。（知识共享署名 2.0）https://www.flickr.com/photos/adam_jones/3773808267/
图 7.10	„Jam Out“ 由 Chris Hunkeler（CC BY-SA 2.0）https://www.flickr.com/photos/chrishunkeler/34127971400/
图 7.18	熊：使用 Imagen 创建（左）：https://arxiv.org/pdf/2205.11487，经许可刊印作者。 莱茵河：使用 Stable Diffusion 创建 https://stablediffusionweb.com/：自行生成的图像，使用自己的图像描述采用 CreativeML Open RAIL-M 许可 https://stablediffusionweb.com/license
图 7.19	“Chianocco-Lesna” by 费德里科 Feroldi https://www.flickr.com/photos/federicoferoldifoto/8403374848, 已修改。CC BY-SA 2.0 许可协议
图 7.20	Taryn 的“Sizzle”https://www.flickr.com/photos/tarale/6689019875 CC BY-SA 2.0 许可协议
图 7.21	“正在进行篮球比赛的球员” Pexel 可免费使用。 https://www.pexels.com/video/guys-playing-a-basketball-game-5275203/ 正在扣篮的男人。Pexel 可免费使用。 https://www.pexels.com/video/man-doing-a-basketball-dunk-5275078/
图 7.22	足球：“Chianocco-Lesna”，作者 Federico Feroldi https://www.flickr.com/photos/federicoferoldifoto/8403374848, 已修改。CC BY-SA 2.0 许可协议。 水牛：Henry Stober 的《The Big 5》视频中的帧。CC BY-SA 3.0 许可协议 https://vimeo.com/239953264
图 7.23	Dog: “小狗” by Jonathan Kriz 裁剪 https://www.flickr.com/photos/27587002@N07/5170590074/in/photolist 许可协议 知识共享署名 2.0 猫：“Day25 Kimba”，作者 Rachel Hofton，已裁剪https://www.flickr.com/photos/rachels_照片_world/3238883214/in/photolist 许可协议 知识共享署名 2.0
图 7.32	Atari Seaquest：来自公共领域 Stella 模拟器的自制快照https://stella-emu.github.io/ Dog: “小狗” by Jonathan Kriz 裁剪 https://www.flickr.com/photos/27587002@N07/5170590074/in/photolist 许可 知识共享署名 2.0
图 7.22	足球：“Chianocco-Lesna”，作者 Federico Feroldi https://www.flickr.com/photos/federicoferoldifoto/8403374848 CC BY-SA 2.0 许可协议 Gnu：《The Big 5》视频中的帧https://vimeo.com/239953264 由 Henry Stober 制作，依据以下许可协议授权：CC BY-SA 3.0 许可协议 https://creativecommons.org/licenses/by-sa/3.0/

表 A.4 第 8 章中使用的图像来源。

图	引用图像
图 8.1	参见表格 A.1 中的图 2.18
图 8.6	“Sophie 玩章鱼衣架”，作者 David Leo Veksler https://www.flickr.com/photos/heroiclife/9872140076 CC BY-SA 2.0 许可协议

索引

一维卷积，13 一维深度卷积，307 3D邻近注意力，350

抽象式摘要，252 ACE，196 ACE 2005数据集，205 动态系统中的动作，355 激活函数，7、95 即席检索，222 AdaDelta优化器，57 AdaGrad优化器，57 Adam优化器，58 Adapter-Bot，263 AdapterHub，136 AdaSpeech 2，312 AE自编码器语言模型，19 仿射变换，6、7 AI人工智能，2 AISO，238 ALBERT，81、186 ALBERT-SEN，185 偶然不确定性，61 Alexa挑战赛，280 AlexNet，13 算法过滤，388 ALIGN，324 与人类偏好对齐，140 Aloe，346 AlphaFold，360 AlphaFold2，381 Amazon 670K数据集，184、187 AmbigQA基准，225 AminoBERT，360、381 类比，10 ANCE，230 AR自回归语言模型，20

人工智能，2 ArXiv基准，184、185、255、257 基于方面的情感分析，207 ASR自动语音识别，278、305 关联分数，22 Atari 基准，356 ATLOP，204 注意力，12、21 注意力头，24 AttentionXML，187 AugZero，266 自编码器，27、51 自编码器语言模型，19 自动语音识别，278、305 自动化风险，392 AutoPrompt，138 自回归语言模型，11、20、37、38、51 AV-ASR，346

BabelNet数据，191、192 回译，108、245 词袋，5、183 装袋法，63 BART，91 批归一化，59 贝叶斯神经网络，61 束搜索，48 BEIR基准，232 BEiT，319 BEM，192 BERT，19、21、184、195 BERTScore指标，50 BertViz，65 偏见，6、383 双向编码器，27 双向RNN，11

BIG-bench 基准，163、
376 BigBird，98、185、
234、255 BigScience 倡议，
86、391 BigSSL 数据，309
双向长短期记忆网络，12 二
元分类，182 必应搜索引擎，
67、240 BioELECTRA，
197 每字节位数，240
BlenderBot 1，281
BlenderBot 2，283
BlenderBot 3，286、389、
391 BLEU 指标，BLEU，
49 BLINK，198 BLOOM，
86、391 布隆过滤器，390
BLURB 基准，197 BoolQ 基
准，160 每字节位数，240
BRIDGE，117 BRIO，254
BriVL，324 ByT5，250 字节
对编码，4

C4 数据集，95 基于图像描述的图像
检索，325 CAST，272 灾难性遗忘，
119，132，133 新闻分类，182 类
别，182 因果自注意力，37 CB 基准，
160 思维链，138 字符 n-gram，4
聊天机器人，278 CheckList 流程，
161，168，172 Chinchilla，87，
347 CIDEr 指标，326 类别，182 分
类损失，6 无分类器重建，331 标题
党内容，384 CLIP，323 CLIP 相似
度得分，327 封闭领域问答系统，
232 闭卷问答，235 完形填空提示，
137，189 完形填空任务，27 CMT，
313 CNN 卷积神经网络，12
CNN/Daily Mail 基准，92，254，
2 COCO 图像描述，322 COCO 数据，
333
coCondenser，231 CogView，127，
329 CoLAKE，118 ColBERT，227
ColTran，320 组合式自监督学习，
308 Combiner，100 常识知识，
166 压缩 Transformer，99
ConceptNet 知识库，268
Conceptual Captions 数据，322，
326 条件随机场，196 确认偏误，
384 Conformer，307 共轭梯度，
58 CoNLL 2003 数据，34
CoNLL05 基准，208
Connectionist 时间分类，308
ConSec，194 上下文向量，12
ContextNet + NST，307 上下文嵌
入，20，23，30，373 对比学习，
323 对比损失，317 控制轨迹，379
卷积层，12 卷积神经网络，12
COOT，342 COPA 基准，160
CORA，249 共指消解，202
CorefQA，202 语料库，5 余弦相似
度，230 CoVeR，345 CQL，356
CRF 条件随机场，196 Crf2o，208
交叉熵，129 交叉注意力，46 跨模态
掩码语言模型，323 CTRL，262

D4RL 基准，356 DALL-E，
327 DALL-E 2，332
Data2vec，321 DBpedia 基
准，111，184 DeBERTa，
83，207 DeBERTaV3，83
DeCEMBERT，343 决策
Transformer，355 解码器，
12，45 解码器块，46

深度学习，2，375 深度神经网络，1，7 DeepCTRL，170 DeepFaceLab，387 DeepProblog，169 DeepSpeed 工具箱，53，58，86，87 稠密检索，223，229 稠密检索器，236 DensePhrases，121 依存句法分析，208 对话系统，278，378 DialogBERT，281 DialoGPT，281 DiffBot，111 差分隐私，390 差分隐私随机梯度下降，390 扩散模型，330 扩散过程，330 膨胀因果卷积，13 狄利克雷分布，62 折扣累计增益，186 解耦注意力，83 解耦嵌入，83 远程监督，209 DistilBERT，130 分布语义学，8 DNA，360，380 DNABERT，360，380 Doc2query，228 DocFormer，211 DocRED 基准，204，206 文档分类，6 DPR 稠密段落检索器，121，230，236，237 DrawBench 数据，333 DropConnect，62 Dropout，59 DropToken，343 药物发现，115，380，381 DSS，102

E-BERT，114 EfficientQA 基准，225 ELECTRA，81 ELI5 基准，241、385 Eliza 聊天机器人，279 ELMo，12 嵌入，2、8、372 上下文，20 上下文化，20 静态，8 段落嵌入，107 编码器，12、44 编码器块，25 实体，182 实体链接，191、198 实体提及，198 EntMask，199 Entmax 变换，41 EntQA，199 Epic-Kitchens-100 数据集，340 认知不确定性，60 ERNIE-Doc，184 ERNIE-THU，114 Escher，193 ESMFold，360 ETC-NLG，263 EURLex-4K 基准，184、187 EWISER，119、192 示例，6 专家系统，375 可解释人工智能，67 扩展 Transformer 架构，98 抽取式摘要，252 极端多标签分类，182、186

Facts as Experts，114 Facts2Story，269 FAISS，193、227、230 假新闻，274 Fast WaveNet，310 FastMoE快速混合专家库，126 FastSpeech 2，310 FastSpeech 2s，311 FastText，10 FB HYBRID，238 FCL全连接层，FCL，7、25 少样本学习，136 少样本提示，349、373 FIDFréchet Inception 距离，327 FiDFusion 解码器，121、237 滤波器内核，12 微调，2、25、28 FIST，270 Flamingo，347 FLAN，141 扁平命名实体识别，195 FLOPS 每秒浮点运算次数，96 Formal，264 前向奖励总和，355 基础模型，3、52、96、371、374 Fréchet Inception 距离，327 Freebase 数据，111 Frozen，326、347 全连接层，7、11、25 FUNSD 基准，211

FVD，350

GAN 生成对抗网络，260，327 间隙句生成，92 门控线性单元，95，307 门控循环单元，11 GATO，357 GauGAN2，328 高斯过程，63 GDC，263 GeDI，262 GEGLU激活，95，128 GEM基准，260 GeneBERT，360，381 泛化误差，59 生成对抗网络，260，327 生成式预训练变换器，37 Genia语料库，197 基因组学，380 GENRE，199 GitHub Copilot，276 GLaM，126 GLIDE，330 GLM，93 全局词元，98 词义 gloss，192 GlossBERT，192 GloVe，10 GLUE数据，33，80 GODIVA，352 Gopher，86，163，170，235 GPT 生成式预训练变换器，20，37 GPT-2，42，260 GPT-3，67，83，98，160，234，256，260，266，271，327 GPT-GNN，115 GPT-J-6B，84 GPT-Neo，85 GPT-NeoX-20B，85，391 GRACE，207 梯度，56 梯度下降，56 梯度噪声，125 图卷积网络，114 Graph-BERT，115 GraphFormers，115 GraphPlan，272 贪心解码，48 GRF，262 语言接地，324 语言落地，288 GSLM，312 GSM8K基准，276 GSM8K-Python基准，276 GSPMD，58 Gumbel-Softmax 分布，307 Gumbel-Softmax松弛，328 GYAFC基准，264

HAHNN，185 HASOC 2019推文数据集，186 HAT，246，255 仇恨言论，185 仇恨言论检测，182 HateXplain 数据集，186 隐藏向量，7 均质化，394 同音异义词，182，223 HotpotQA基准，238 HowTo100M 数据，344 HumanEval基准，276 HYBRIDER，236 HyperGrid，134 超参数，96

独立同分布，6 IE 信息抽取，181 图像标注，324 图像描述，324 图像处理，378 Imagen，333 Imagen Video，ImageNet基准，13 IMDB基准，82，184 Inception 模型，327 Inception 分数，327 归纳偏置，51 信息抽取，181，376 信息检索，222，377 参数初始化，57 创新的，396 输入嵌入，21 InstructGPT，139，261 集成梯度，66 倒排索引，222 IOB2 格式，195 IR 信息检索，222 IS Inception 分数，327

JAX，55 JFT-3B 基准测试，345 JNLPBA 基准测试，196 Jukebox，313 Jurassic-1，86

K-Adapter，119

知识库 知识库，78、110、232
KeBioLM，196 KEPLER，113
Key-to-Door基准，356 键向量，22 基于关键词的检索，222
KGPool，209 KGPT，120
Kinetics数据，340 KL Kullback-Leibler散度，62
KnowBERT，114 知识库，78、110、111、232
Kullback-Leibler散度，62

L1（L_1）正则化，59 L2（L_2）正则化，59 标签，182 标签平滑，59 LaBSE，107 LAFITE，330 Laion数据，322
LAION-5B数据，334 LAMB优化器，58
LAMBADA基准测试，87、88、161、240、261、375 LaMDA，67、240、262、284、380、389 语言识别方法，5 语言模型，11、19 自编码器，19 自回归，11、20 编码器-解码器，20 掩码，26 拉普拉斯近似，62 潜在狄利克雷分配，272 层归一化，25、59、95 LayoutLM3，211
LayoutXLM，211 学习：少样本，136 单样本，136 学习率，56 自监督，8 监督，8 转移，132 无监督，8 零样本，136
Leviated标记，205 词汇替换，191
LibriLight语音数据，312 LibriSpeech基准，307 LightXML，188 似然，61 李克特量表，256 LIME，65
线性变换，6 线性Transformer，100 LJ Speech数据，313 LM语言模型，19 局部最小值，56 对数似然，38 逻辑分类器，6、10、183 Long Range Arena基准，100、102 长短期记忆，11 Longformer，98、256 LoRA，135 损失函数，79
LSA，207 LUKE，119、195

M2M，246 Macaw，234 机器学习，2，375 机器翻译，12 无监督，108
MAD-X，135 MaMa，208 MAML，134 平均精度均值，225 MARGE，108 间隔，7 马尔可夫链蒙特卡洛，61 掩码语言模型，26 掩码语言建模，316 掩码多头自注意力，45 掩码区域预测，323 掩码自注意力，37 MASS，90 大规模多任务语言理解基准，162 最大池化，12 最大熵损失，6 最大似然估计，38 最大似然原理，6 mBART，108
mBERT，104 MBPP基准，276 平均精度均值，225 平均意见得分，310 平均倒数排名，114，225 Meena，281
Megatron-CNTRL，268
Megatron-LM，85 梅尔频率倒谱系数，306 MeMViT，344 提及，191，195
Merlot，346 元学习，134 METEOR指标，METEOR，50

梅尔频率倒谱系数（MFCC），306 小批量，56 Softmax混合，41、95 混合专家模型，96、107、126、127 MKQA基准，249 ML 机器学习，2 MLE 最大似然估计，38 MLM，26 MLP 多层感知器，7 MLSum基准，254 MNEMELM，271 MobileBERT，130 模型，6 模型卡，174 模型预测控制，400 模型量化，128 模型 1，228 MoDNA，381 混合专家 混合专家模型，107、126 Moments in Time数据，340 动量优化器，57 monoBERT，226 monoT5，227 蒙特卡洛近似，61 MOS 平均意见分，310 MPG，115 MRC，196 MRR 平均倒数排名，114、225 MS-MARCO数据，224 MS UnitedQA，238 MT-NLG，87 mT5，107、250 mT6，107 MTR，238 MTV，345 MUDERN，283 MuLaN，192 多文档摘要器，253 多头交叉注意力，46 多头自注意力，25 多示例学习，209 Multi-News 基准，257 多会话聊天，283 多流架构，316 多类分类，182 多标签分类，182 多层感知器，7 多层循环神经网络，12 多模态内容分析，181 多模态问答，232 MultiNLU基准，109 MultiRC基准，160 MuseNet，313 Music Transformer，313

n元语法，4 NÜWA，337，350 朴素贝叶斯，183 命名实体，166，182，195 命名实体识别，29，34，182，195 NAS 神经架构搜索，60 自然语言生成，181，258 推理，172 处理，1 理解，32，159 Natural Questions 基准，120，121，225，233，237 朴素实在论，384 NE 命名实体，166 近邻搜索，121 否定，165 NER 命名实体识别，29，34，182，195 嵌套实体，196 神经架构搜索，60 《纽约时报》数据集，206，210 下一句预测，27 NLG 自然语言生成，181 NLI 自然语言推理，172 NLP 自然语言处理，1 NLU自然语言理解，32，159 node2vec，114 噪声对比估计，9 NQ Natural Questions 基准，225，233

目标检测，324 OFA，335 Omnivore，344 一次性学习，136 OntoNotes基准，202，208 OntoNotes共指数据，81 开放域问答系统，232 开放域问答，120 开放多语言WordNet，191 开卷问答，236 OpenGPT-X，391 OPT，86，286，391 OPTIMUS，265 OSCAR，325 过拟合，59，124，131 Overlap，188

PaLM，88、160、164、170、235、248、250、254；PaLM-Coder，88；PanGu-α，86；Parabel，187

释义，167、222；主成分分析，39；PEGASUS，92、253；宾夕法尼亚树库语料库，39、42；Perceiver，100；Perceiver AR，101；Perceiver IO，101；感知器重采样器，347；感知损失，320；Performer，99；困惑度，39；角色设定，282；PET，119、189；Pile 数据集，85、127、240、337；PL-Marker，205；PLATO，397；Plato-2，282；预训练语言模型，2、51；PlotMachines，268；Pointer，268；策略，355；PoolingFormer，234；位置嵌入，21；后验分布，61；PPLM，262；预训练语言模型，2、51；预训练，2、25；k 处的精确率，186；前缀提示，137；Primer，93、256；主成分分析，39；先验分布，61；概率软逻辑，169；产品键内存，96；乘积键，82；ProGen，269；ProGeT，269；提示，136、373；提示设计，137；宣传，384；蛋白质，381；蛋白质组学，380；近端策略优化，140；概率软逻辑，169；PubLayNet 基准，211；PubMed 语料库，196、255；PubMedBERT，197；PyTorch，55

Q-BERT，128 拟牛顿法，58 查询，222 查询流，82 查询向量，22

问答，29、232、377 开放域，120

RACE 数据集，80；RAFT 基准，189；检索增强生成，237；随机森林，183；随机采样，40；RankNAS，60；Re-TACRED 基准，203；阅读器，121、236、237；阅读理解，232；RealFormer，234；REALM，237；REBEL，206；RECENT，203；推荐系统，115；ReCoRD 基准，160；修正线性单元，7；循环神经网络，10；RedCaps 数据集，322；RefCOCO 基准，336；Reformer，99；区域描述，343；正则化方法，59；强化学习，140、355；人类反馈强化学习，140；关系，111；关系抽取，182、376；Relation-QA，203；ReLIE，210；修正线性单元，7；RemBERT，106；替换词元检测，81；自然语言处理的可复现性，174；残差连接，13、25、58；ResNet，13；检索，196；检索器，121、236；检索器-阅读器架构，121；Retro，170、239、261；动态系统中的奖励，355；奖励模型，140；RL-175B，255；RMS归一化，95；RMSProp优化器，57；循环神经网络，10；RoBERTa，80、186、203；RocketQA，231；RoseTTAFold，360；ROUGE 指标，ROUGE，49；路由网络，126；RoutingTransformer，99；RPT，117

RTE基准，160 基于规则，279

S4，101、307；鞍点，56；安全性，386；显著性，65；显著跨度掩码，92；采样方法，61；讽刺作品，384；SayCan，401；缩放点积注意力，23；SCaNN库，239；得分向量，6；搜索引擎，222；Segatron，99；自注意力，23、373；词元自相似性，40；自监督学习，372；自监督训练，8；语义知识，166；语义解析，117；SemCor数据，193；SemCor3.0数据，192、194；SemEval 2014任务4.2基准，207；SemEval-13数据，193；SemEval-15数据，193；SemEval-20任务12基准，184；Senseval-3 数据集，31；合理性，386；句子嵌入，107；SentenceBERT，229；SentencePiece，5；情感分析，182、183；Seq2seq序列到序列，44；序列到序列模型，44；SGD随机梯度下降，56；SHAP，65；简单循环神经网络，11；SimVLM，326；单文档摘要器，253；单流架构，316；SMAN，275；SMITH，228、234；SNGP，62；SNLI-VE数据，336；SOE，270；softmax 函数，6；Something-Something V2数据，340；最先进水平，3；SOTA最先进水平，Sota，3；spaCy工具包，4；垃圾邮件检测，182；片段预测，29

SpanBERT，80，202，203 Sparse Transformer，98，328 特异性，386 语音识别，378 SpeechStew，309 球面嵌入，10 SQA数据，117 SQuAD 1.0数据，34，80 SQuAD 2.0数据，80，232 SSAN，204 ST-MoE-32B，128，161，174，233，254 Starspace，10 动态系统中的状态，355 最先进水平Sota，3 静态嵌入，8 统计披露，390 刻板印象，383 STIE，255 差分隐私随机梯度下降，56 差分隐私随机梯度下降，390 随机鹦鹉，396 故事生成，377 StrategyQA基准，170 StructBERT，81 结构不确定性，60 学生模型，129 StyleGAN2，330 StyleLM，265 StyleSwin，321 摘要生成，252 摘要器，252 总结，252 抽象式，252 抽取式，252 SUN RGB-D 数据集，344 超分辨率，320 SuperGLUE 基准，83，95，127，138，159 监督训练，8 支持向量机，7，183 SwiGLU激活函数，95 Swin Transformer，320，321，344 SwinIR，320 Swish激活函数，192，307 Switch，96，107，127 符号落地，324，339 同义词，222 同义词集，191，192 句法知识，165 Synthesizer，95，100 系统1（快速思考），399 系统2（慢速思考），400

t-SNE投影，31

T5，92 T5-XXL，174 TaBERT，116 TableGPT，117 TACKBP-2010基准测试，199 Tacotron 2，310 TACRED 基准，113 Tapas，117 任务适配器，135 教师强制，38 教师模型，129 TeKGen，119 张量，5 TensorFlow，55 TERA，309 测试数据，6 文本分类，29，182 文本对分类，29 文本摘要，377 文本转语音，278，310 TF-IDF 统计量，5 主题 Transformer，313 思维链，67，170，235 时间序列预测，102 TinyBERT，130 太长不看基准测试，254 TLDR，255 TLM，105 词元，4，19 词元嵌入，21 TokenSwitch，128 Top-k 准确率，225 Top-k 采样，41 Top-p 采样，41 有害语言，383 训练语料库，5 数据，6 自监督，8 监督，8 无监督，8 TransD，112 TransE 模型，112 TransE 损失，112 迁移学习，2，25，28，132，372 Transformer，20，44，51 编码器-解码器，20，51 TransformerTTS，310 Transformer-LS，100 Transformer-XL，99 TransH，112 翻译，377 翻译语言建模，105 TriviaQA基准测试，120，122，233 真值偏差，384 TruthfulQA 基准，233 TTS 文本转语音，278，310 图灵测试，259 Turing-NLG，87 TURL，116 TyDiQA-GoldP 基准，249

U-Net，331 UDG 无监督数据生成，142 欠规范性，133 Unicoder，106 单向编码器，38 UniLM2，93 UniRE，204 无监督数据生成，142 无监督机器翻译，108 无监督训练，8

VAE变分自编码器，260，327 值向量，22 变分自编码器，260，327 变分推断，62 变分下界，330 VATT，343 向量量化变分自编码器，329 视频图像描述，342 视频图像块，340 视频处理，378 VideoTransformer，350 VideoBERT，342 VilBERT，324 VinVL，325 虚拟助手，278 VisionTransformer，317 视觉蕴含任务，336 视觉问答，325 视觉推理，346 视觉词元，322 ViT VisionTransformer，317 VIVO，326 词汇表，5 词汇不匹配问题，223 体素，343 VQ-GAN，320 VQ-VAE 向量量化变分自编码器，319，329 VQA v2 基准，326 VQA 视觉问答，325

w2v-BERT，308；
wav2vec 2.0，307；
WaveGlow，313

WaveNet，13、310；WebGPT，67、240；WebQuestions基准测试，95；WER 词错误率，305；WiC 基准，160；Wiki65K 基准，228；Wikidata 知识库，119；Wikidata5M 知识库，118；WikiHop基准测试，234；维基百科，191；WikiTableQuestions基准测试，116；WikiText-103 基准，86、161；Winograd基准测试，173；Winogrande 基准，173；WIT 数据，323；WKLM，118；WMT2014 英德翻译基准，95；WMT2019 德英基准测试，108；Woebot，380；Word n-gram，4；词语标注，29、192；词嵌入，8；词错误率，305；词义，166；词义消歧，182、191；Word2Vec，9

WordNet 数据，111、191、192；WordNet Glos 数据，193；WordPiece，5；工作记忆，400；世界知识，166；世界模型，401；WSC 基准，160；词义消歧，182、191；悟道 2.0，126、337

小冰，280、389；XLM，105；XLM-R，106；XLM-R_{XXL}，106；XLNet，82、184、185；XLNG，108；XMC-GAN，328；XNLI 基准，83、104、107；XSum 基准，95、254；Xtreme 基准，107

YAGO，111；Yelp 基准测试，184

零样本学习，42、136