基于 GPT-3、ChatGPT、GPT-4 等 Transformer 架构的自然语言处理

[法]丹尼斯·罗斯曼(Denis Rothman) 著

叶伟民 译

清華大學出版社

北 京

北京市版权局著作权合同登记号　图字：01-2023-1361

图书在版编目(CIP)数据

基于 GPT-3、ChatGPT、GPT-4 等 Transformer 架构的自然语言处理/(法)丹尼斯・罗斯曼著；叶伟民译. —北京：清华大学出版社，2024.1

书名原文：Transformers for Natural Language Processing: Build, Train, and Fine-tune Deep Neural Network Architectures for NLP with Python, Hugging Face, and OpenAI's GPT-3, ChatGPT, and GPT-4, Second Editon

ISBN 978-7-302-64872-7

I. ①基…　II. ①丹…　②叶…　III. ①人工智能－应用－自然语言处理－研究　IV. ①TP391

中国国家版本馆 CIP 数据核字(2023)第 215136 号

责任编辑：王　军　韩宏志
封面设计：孔祥峰
版式设计：思创景点
责任校对：马遥遥
责任印制：丛怀宇

出版发行：清华大学出版社
网　　址：https://www.tup.com.cn，https://www.wqxuetang.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-83470000　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京同文印刷有限责任公司
经　　销：全国新华书店
开　　本：170mm×240mm　　**印　　张**：23.5　　**字　　数**：514 千字
版　　次：2024 年 1 月第 1 版　　**印　　次**：2024 年 1 月第 1 次印刷
定　　价：99.80 元

产品编号：100097-01

专家推荐

在不到 4 年的时间里，Transformer 模型以其强大的性能和创新的思想，迅速在 NLP 社区崭露头角，打破了过去 30 年的记录。BERT、T5 和 GPT 等模型现在已成为计算机视觉、语音识别、翻译、蛋白质测序、编码等各个领域中新应用的基础构件。因此，斯坦福大学最近提出了“基础模型”这个术语，用于定义基于巨型预训练 Transformer 的一系列大型语言模型。所有这些进步都归功于一些简单的想法。

本书可作为所有对 Transformer 工作原理感兴趣的人的参考书。作者在理论和实践两方面都做出了出色的工作，详细解释了如何逐步使用 Transformer。阅读完本书后，你将能使用这一最先进的技术集合来增强你的深度学习应用能力。本书在详细介绍 BERT、RoBERTa、T5 和 GPT-3 等流行模型前，先讲述了 Transformer 的架构以便为你的学习奠定坚实基础。本书还讲述了如何将 Transformer 应用于许多用例，如文本摘要、图像标注、问答、情感分析和假新闻分析等。

如果你对这些主题感兴趣，那么本书绝对是值得一读的。

——Antonio Gulli
Google 工程总监

作者简介

Denis Rothman 毕业于法国巴黎索邦大学和狄德罗大学，设计了首批获得专利的编码和嵌入系统，编写了首批获得专利的 AI 认知机器人和机器人。他的职业生涯始于为 Moët et Chandon 提供 NLP(自然语言处理)聊天机器人，并为空中客车公司(前身为 Aerospatiale)提供 AI 战术防御优化器。此后，Denis 为 IBM 和奢侈品牌开发了 AI 资源优化器，并最终发展为在全球范围内使用的 APS(高级规划和调度)解决方案。

致谢

我要感谢那些从一开始就信任我并分担持续创新风险的公司。还要感谢我的家人，他们一直相信我会成功。

审校者简介

George Mihaila 是北得克萨斯大学计算机科学系的在读博士生，他也在该校获得了计算机科学硕士学位。他在自己的祖国罗马尼亚获得了电气工程学士学位。

George 在 TCF 银行工作了 10 个月，在那里帮助建立了自动模型部署和监控的机器学习操作框架。他在 State Farm 实习期间担任数据科学家和机器学习工程师，在北得克萨斯大学高性能计算中心担任数据科学家和机器学习工程师。他在 NLP 领域工作了 5 年，其中最后 3 年使用了 Transformer 模型。他的研究兴趣是个性化对话生成。

George 是本书第 1 版的技术审校者。

George 目前正在攻读博士学位。

空闲时间 George 喜欢通过教程和文章分享他对最先进语言模型的理解，并帮助 NLP 领域的其他研究人员。

前　言

Transformer 是自然语言理解(Natural Language Understanding，NLU)的游戏规则改变者，NLU 是自然语言处理(Natural Language Processing，NLP)的一个子集。NLU 已成为全球数字经济中 AI 的支柱之一。

Transformer 模型标志着 AI 新时代的开始。语言基础已成为语言建模、聊天机器人、个人助理、问答、文本摘要、语音转文本、情绪分析、机器翻译等的支柱。社交网络正在取代实体接触，电子商务正在取代实体购物，数字报纸、流媒体正在取代实体剧院，远程文档咨询正在取代实体访问，远程工作正在取代现场办公，我们正在见证数百个领域的类似趋势。如果没有理解 AI 语言，社会上使用网络浏览器、流媒体服务和任何涉及语言的数字活动都将非常困难。我们的社会从物理信息到海量数字信息的范式转变迫使 AI 进入一个新时代。AI 已经发展到数十亿级参数模型，以应对万亿级单词数据集的挑战。

Transformer 架构具有革命性和颠覆性，它打破了过往 RNN 和 CNN 的主导地位。BERT 和 GPT 模型放弃了循环网络层，使用自注意力机制取而代之。Transformer 模型优于 RNN 和 CNN。这是 AI 历史上划时代的重大变化。

Transformer 编码器和解码器包含单独训练的注意力头(attention head)，并能使用 GPU、TPU 等尖端硬件进行并行化。注意力头可以使用 GPU 运行，从而为十亿级参数模型和即将出现的万亿级参数模型打开大门。OpenAI 在一台具有 10 000 个 GPU 和 285 000 个 CPU 内核的超级计算机上训练出具有 1750 亿个参数的 GPT-3 Transformer 模型。

随着数据量的不断增长，训练 AI 模型需要的规模也越来越大。Transformer 模型为参数驱动的 AI 开启了新时代。我们需要大量参数进行学习，才能学习到由数以亿计的单词组合的数据集。

Google BERT 和 OpenAI GPT-3 等 Transformer 模型将 AI 提升到另一个层次。Transformer 可以执行数百项它们没有接受过训练的 NLP 任务。

Transformer 还可通过将图像视为单词序列来学习图像分类和重构图像。本书将介绍尖端的计算机视觉 Transformer，如 Vision Transformer(ViT)、CLIP 和 DALL-E。

基础模型是指经过充分训练的、不需要微调即可执行数百项任务的 Transformer 模型。这种规模的基础模型是我们在这个海量信息时代所需的工具。

想想每天需要多少人来控制社交网络上发布的数十亿条消息的内容，以便在提取所包含的信息之前确定是否合法和合乎道德。

想想每天在网络上发布的数百万页文字需要多少人来翻译。或者想象一下，如果要人工对每分钟多达数百万条消息进行控制需要多少人力资源！

想想将每天在网络上发布的所有大量流媒体转换为文字需要多少人力资源。想想为不断出现的数十亿幅在线图像生成 AI 图像字幕需要多少人力资源。

本书将带领你开发代码和设计提示(这是一项控制 Transformer 模型行为的新的“编程”技能)。每一章都会使用 Python、PyTorch 和 TensorFlow 从头开始讲授语言理解的关键方面。

你将学习原始 Transformer、Google BERT、OpenAI GPT-3、T5 和其他几个模型的架构。最后一章将在前面 16 章所学知识的基础上，展示 ChatGPT 和 GPT-4 的增强能力。你将学会如何微调 Transformer，如何从头开始训练模型，如何使用强大的 API。Facebook、Google、Microsoft 和其他大型科技公司提供了大量数据集供我们探索。

你会密切关注市场上对语言理解的需求，例如媒体、社交媒体和研究论文等领域。在数百项 AI 任务中，我们需要总结大量的研究数据，为各个领域翻译文件，并出于伦理和法律原因扫描所有社交媒体帖子。

整本书将使用 Python、PyTorch 和 TensorFlow 进行实战。你将首先学习 AI 语言理解神经网络模型的要素，然后学习如何探索和实现 Transformer。

本书旨在为读者提供在这个颠覆性的 AI 时代中，有效开发语言理解关键方面所需的 Python 深度学习知识和工具，呈现成为工业 4.0 AI 专家所需要的新技能。

本书读者对象

本书并不介绍 Python 编程或机器学习概念，而是专注于机器学习的机器翻译、语音到文本、文本到语音、语言建模、问答和更多 NLP 领域。

本书读者对象包括：

- 熟悉 Python 编程的深度学习和 NLP 从业者。
- 数据分析师和数据科学家，他们希望了解 AI 语言理解，从而完成越来越多的语言驱动的功能。

本书内容

第 1 章“Transformer 模型介绍”从较高层次解释什么是 Transformer 模型。我们将研究 Transformer 生态系统和基础模型的特性。该章重点介绍许多可用的平台以及工业 4.0 AI 专家的发展历程。

第 2 章“Transformer 模型架构入门”通过回顾 NLP 的背景，讲述了循环神经网络(RNN)、长短期记忆网络(LSTM)和卷积神经网络(CNN)深度学习架构是如何演变为

Transformer 架构的。我们将通过 Google Research 和 Google Brain 的作者们独创的“注意力机制就是一切(Attention Is All You Need)”的方法来分析 Transformer 的架构。将描述 Transformer 的理论，并通过 Python 实践来讲解多头注意力子层是如何工作的。通过本章的学习，你将理解 Transformer 的原始架构，从而为后续章节探索 Transformer 多种变体和用法打下良好基础。

第 3 章“微调 BERT 模型”基于原始 Transformer 的架构进行扩展。BERT(Bidirectional Encoder Representations from Transformers)向你展示了一种理解 NLP 世界的新方式。与通过分析过去序列来预测未来序列不同，BERT 关注整个序列！首先介绍 BERT 架构的关键创新，然后通过在 Google Colab 笔记本中逐步执行每个步骤来微调一个 BERT 模型。与人类一样，BERT 可以学习任务并执行其他新任务，而不需要从头学习。

第 4 章“从头开始预训练 RoBERTa 模型”使用 Hugging Face PyTorch 模块从头构建一个 RoBERTa Transformer 模型。这个 Transformer 模型既类似于 BERT，又类似于 DistilBERT。首先，我们将使用自定义数据集从头训练一个词元分析器。然后将使用训练好的 Transformer 运行下游的掩码语言建模任务。

第 5 章“使用 Transformer 处理下游 NLP 任务”揭示了 Transformer 模型在下游 NLP 任务中的神奇之处。我们可以微调预训练 Transformer 模型以执行一系列 NLP 任务，如 BoolQ、CB、MultiRC、RTE、WiC 等在 GLUE 和 SuperGLUE 排行榜上占据主导地位的 NLP 任务。将介绍 Transformer 的评估过程、任务、数据集和评估指标。然后将使用 Hugging Face 的 Transformer 流水线处理一些下游任务。

第 6 章“机器翻译”讲述什么是机器翻译，并讨论如何从依赖人类翻译的基准转向使用机器翻译的方法，从而帮助读者理解如何构建机器翻译系统并进行进一步的研究和开发。然后，我们将预处理来自欧洲议会的 WMT 法英数据集。机器翻译需要精确的评估方法，这一章将讲述 BLEU 评分方法。最后，我们将使用 Trax 实现一个 Transformer 机器翻译模型。

第 7 章“GPT-3”探索了 OpenAI GPT-2 和 GPT-3 Transformer 的许多方面。首先研究 OpenAI GPT 模型的架构，解释 GPT-3 引擎。然后将运行一个 GPT-2 345M 参数模型，并与之交互生成文本。接着将讲述 GPT-3 playground 的实际应用，使用 GPT-3 模型运行 NLP 任务，并将结果与 GPT-2 进行比较。

第 8 章“文本摘要(以法律和财务文档为例)”介绍 T5 Transformer 模型的概念和架构。我们将使用 Hugging Face 初始化一个 T5 模型进行文本摘要。将使用 T5 模型汇总各种文本，然后探索应用于 Transformer 的迁移学习方法的优点和局限性。最后，将使用 GPT-3 将一些公司法律文本汇总为小学二年级学生都能看懂的文本。

第 9 章“数据集预处理和词元分析器”分析词元分析器的局限性，并介绍一些改进数据编码过程质量的方法。首先构建一个 Python 程序，调查为什么一些单词会被 Word2Vector 词元分析器省略或误解，讲述预训练词元分析器的局限性。然后我们改进了第 8 章 T5 模型生成的摘要，以展示词元化过程方法仍然有很大的改进空间。最

后，将测试 GPT-3 语言理解能力的极限。

第 10 章“基于 BERT 的语义角色标注”探索 Transformer 如何学习理解文本内容。语义角色标注(SRL)对人类来说是一项具有挑战性的任务。Transformer 能够产生令人惊讶的结果。我们将使用 Google Colab 笔记本实现由 Allen AI 研究所设计的基于 BERT 的 Transformer 模型。还将使用该研究所的在线资源来可视化 SRL 的输出。最后将讲述 SRL 的局限性和适用范围。

第 11 章“使用 Transformer 进行问答”展示 Transformer 如何学习推理。Transformer 能够理解文本、故事，并进行推理。我们将看到如何通过添加 NER 和 SRL 来增强问答过程。我们将介绍如何设计并实现一个问题生成器；它可以用于训练 Transformer 模型，也可以单独使用来生成问题。

第 12 章“情绪分析”展示了 Transformer 如何改进情绪分析。我们将使用斯坦福情绪树库对复杂句子进行分析，然后挑战几个 Transformer 模型，看看是否能够理解序列的结构及其逻辑形式。我们将看到如何使用 Transformer 进行预测，并根据情绪分析的输出触发不同的行为。该章最后还列举一些使用 GPT-3 的案例。

第 13 章“使用 Transformer 分析假新闻”深入讲述假新闻这个热门话题，以及 Transformer 如何帮助我们理解每天在网络上看到的在线内容的不同观点。每天有数十亿条消息、帖子和文章通过社交媒体、网站和各种实时通信方式发布在网络上。我们将利用前几章介绍的技术来分析关于气候变化和枪支管控的辩论。我们将讨论在合理怀疑的基础上如何确定什么可以被视为假新闻，以及什么新闻仍然是主观的道德和伦理问题。

第 14 章“可解释 AI”通过可视化 Transformer 模型的活动来揭开 Transformer 模型的面纱。我们将使用 BertViz 来可视化注意力头，并使用语言可解释性工具(LIT)进行主成分分析(PCA)。最后将使用 LIME 通过字典学习来可视化 Transformer。

第 15 章“从 NLP 到计算机视觉”深入研究高级模型 Reformer 和 DeBERTa，并使用 Hugging Face 运行示例。Transformer 可将图像视作单词序列进行处理。该章还将研究各种视觉 Transformer 模型，如 ViT、CLIP 和 DALL-E；我们将使用计算机视觉任务测试它们，包括图像生成。

第 16 章“AI 助理”讲述了当工业 4.0(I4.0)达到成熟阶段时，我们将主要与 AI 助理(Copilot)一起工作。AI 助理主要基于提示工程，所以该章首先列举几个非正式/正式英语提示工程的示例，使用 GitHub Copilot 来辅助生成代码。然后讲述视觉 Transformer 如何帮助 NLP Transformer 可视化周围的世界。最后将创建一个基于 Transformer 的推荐系统，可将它应用于数字人和元宇宙中！

第 17 章“ChatGPT 和 GPT-4”在前几章的基础上，探索了 OpenAI 最先进的 Transformer 模型 ChatGPT 和 GPT-4。将使用 ChatGPT 建立对话式 AI，并学习如何使用可解释 AI 解释 Transformer 的输出。将探索 GPT-4，并使用提示编写一个 k-means 聚类程序。还将介绍一个高级用例。最后将使用 DALL-E 2 来创建和生成图像的变体。

附录 A“Transformer 模型术语”讲述 Transformer 的高层结构(从堆叠和子层到注意力头)。

附录 B“Transformer 模型的硬件约束”比较了使用 CPU 和 GPU 运行 Transformer 的性能。我们将看到为什么 Transformer 和 GPU 是完美的绝配，并通过使用 Google Colab CPU、Google Colab 免费版 GPU 和 Google Colab 专业版 GPU 来测试得出的结论。

附录 C“使用 GPT-2 进行文本补全”详细讲述如何使用第 7 章讲述的 GPT-2 进行通用文本补全。

附录 D“使用自定义数据集训练 GPT-2 模型”补充了第 7 章的内容，通过使用自定义数据集构建和训练一个 GPT-2 模型，并使用自定义文本与其进行交互。

附录 E“练习题答案”提供每章末尾练习题的答案。

如何阅读本书

本书大部分程序都使用 Google Colab 笔记本。你只需要一个免费的 Google Gmail 账户，就可以使用 Google Colab 的免费 VM 运行这些笔记本。不过对于某些教学性程序，你需要在你的计算机上安装 Python 来运行。

请花时间阅读第 2 章和附录 A。第 2 章讲述了原始 Transformer，该模型是使用附录 A 讲述的构建模块构建而成的，第 2 章和附录 A 的这些基础知识将在整本书都会用到。如果你觉得这些基础知识难以理解，可以先阅读后面的章节。当通过阅读后续章节对 Transformer 更加熟悉后，再回头阅读第 2 章。

可以在阅读每章后，考虑如何为客户实现 Transformer，或者如何利用它们的新颖机理在你的职业生涯中取得进步。

在线资源

本书的代码、附录 A～D、各章练习题的答案(附录 E)等在线资源，可通过扫描本书封底的二维码下载。另外，读者可扫描封底二维码来下载彩图。

参考资料

将各章的参考资料汇集在一个文档中，读者可扫描封底二维码来下载该文档。

目　录

第 1 章

Transformer模型介绍

Transformer 是工业化、同质化的后深度学习模型，其设计目标是能够在高性能计算机(超级计算机)上以并行方式进行计算。通过同质化，一个 Transformer 模型可以执行各种任务，而不需要微调。Transformer 使用数十亿参数在数十亿条原始未标注数据上进行自监督学习。

这些后深度学习架构称为基础模型。基础模型 Transformer 是始于 2015 年的第四次工业革命的一部分(通过机器-机器自动化将万物互联)。工业 4.0(I4.0)的 AI，特别是自然语言处理(NLP)已经远远超越了过往时代，颠覆了以往的开发范式。

在不到五年的时间里，AI 已经通过高效的云服务提供 API，将大量用户无缝衔接到其各自的软件系统中。许多情况下，通过下载库进行开发这种老旧范式只适用于教学性练习。

工业 4.0 的项目经理通过访问 OpenAI 的云平台、注册、获取 API 密钥，只需要几分钟就能开始工作。用户可以马上输入文本，指定 NLP 任务，并获得 GPT-3 Transformer 引擎发送的回答。通过 GPT-3 Codex，用户不需要先学习大量的编程知识就能编写应用程序。并因此诞生了一项基于 Transformer 模型的新技能——提示工程。

但是，有时 GPT-3 模型可能不适合特定任务。这时候项目经理、顾问或程序员可能希望使用 Google AI、Amazon Web Services(AWS)、Allen AI 研究所或 Hugging Face 提供的其他系统。

项目经理应该选择在本地实施，还是应该直接在 Google Cloud、Microsoft Azure 或 AWS 上实施？开发团队应该选择 Hugging Face、Google Trax、OpenAI 还是 AllenNLP？AI 专家或数据科学家应该使用不需要额外 AI 开发工作的 API 吗？

答案是你需要考虑以上所有选项并有所准备。因为你不知道未来的雇主、客户或用户可能想要或指定什么。因此，你必须做好准备以适应将来出现的任何需求。本书并未讲述市场上存在的所有解决方案。但是，我认为本书为读者提供了足够的解决方案以适应工业 4.0 时代 AI 驱动的 NLP 挑战。

本章先从较高层面解释什么是 Transformer，然后解释灵活理解 Transformer 各种实现方法的重要性。市场上提供的 API 和自动化数量众多，使得平台、框架、库和语言

的定义变得模糊不清。

最后，本章介绍了在 AI 助理(Copilot)技术不断进步的背景下，工业 4.0 AI 专家的技能要求。

在开始探索本书中描述的各种 Transformer 模型实现之前，我们需要先讲清楚一些关键概念。

本章涵盖以下主题：

- 第四次工业革命(工业 4.0)
- 基础模型的范式变革
- 一项新技能——提示工程
- Transformer 的背景
- 实施 Transformer 的难点
- 颠覆性的 Transformer 模型 API
- 选择 Transformer 库的难点
- 选择 Transformer 模型的难点
- 工业 4.0 AI 专家的技能要求
- AI 助理

我们首先讲述 Transformer 的生态系统。

1.1 Transformer 的生态系统

Transformer 模型带来一种崭新的范式变化，以至于需要一个新名称来描述：基础模型。斯坦福大学为此创建了基础模型研究中心(CRFM)。2021 年 8 月，CRFM 发表了一篇由100 多名科学家和专业人士撰写的200多页的论文(详见本书末尾“参考资料”)：On the Opportunities and Risks of Foundation Models。

基础模型并非由学术界创建，而是由大型科技公司创建。例如，Google 发明了 Transformer 模型，从而推出了 Google BERT。Microsoft 与 OpenAI 合作开发了 GPT-3。

大型科技公司不得不找到更好的模型来应对流入数据中心的 PB 级数据的指数增长。这就是 Transformer 模型的诞生背景。

接下来先讲解工业 4.0，以了解为什么需要工业化 AI 模型。

1.1.1 工业 4.0

第一次工业革命实现了机械化。第二次工业革命催生了电力、电话和飞机。第三次工业革命(信息技术革命)带来了数字化。

第四次工业革命(或称工业 4.0)催生了万物互联：机器、机器人、可连接设备、自动驾驶汽车、智能手机、通过社交媒体收集数据的爬虫等。

数百万台机器和机器人每天生成数十亿条数据记录：图像、声音、文字和事件，如图 1.1 所示。

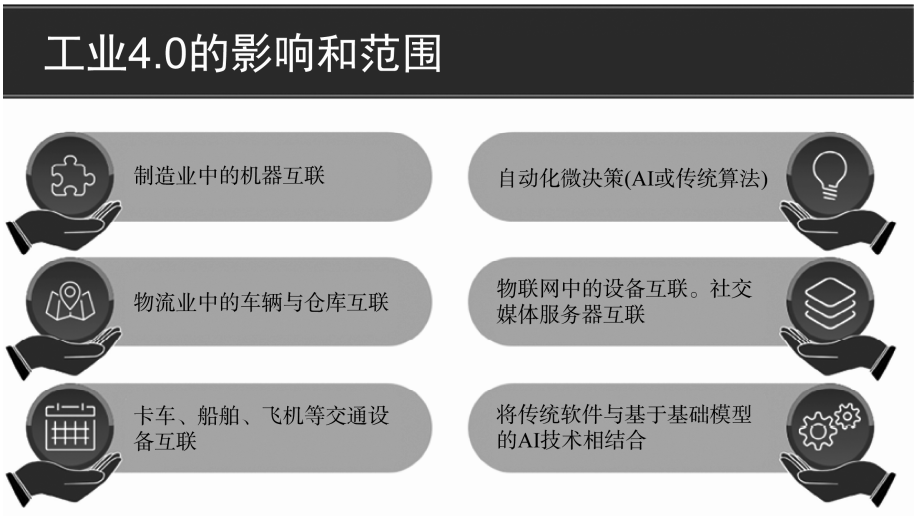

图 1.1　工业 4.0 的范围

因此工业 4.0 依靠不需要大规模人工干预的智能算法来处理数据并做出决策，以应对规模空前的数据量。

因此大型科技公司需要一个 AI 模型就能处理各种任务，而这些任务过往是需要用不同的算法单独处理的。这些就是 Transformer 模型的诞生背景。

1.1.2　基础模型

Transformer 模型有两个显著特点：高度的同质化和令人惊叹的新特性。同质化使得可以使用一个模型来执行各种各样的任务。新特性改变了 AI 解决任务的方式，变成了先在超级计算机上训练 10 亿参数的基础模型，然后去发掘基础模型的能力和应用。

这种范式变革令基础模型成为如图 1.2 所示的后深度学习生态系统的基础核心。

基础模型虽然采用了创新架构，但它们是建立在传统 AI 基础之上的。因此，AI 专家的技能要求越来越广！

当前的 Transformer 模型生态系统不同于 AI 的以往任何演变，可以总结为四个特点。

- 模型架构

Transformer 模型是工业级的。模型的层是相同的，并且专门针对并行处理进行设计。我们将在第 2 章详细介绍 Transformer 的架构。

- 数据

大型科技公司拥有人类历史上最庞大的数据源。数据起源于第三次工业革命(信息技术革命)，并在工业 4.0 的推动下扩大到令人难以想象的规模。

- 计算能力

大型科技公司拥有前所未有的计算能力。例如，GPT-3 的训练速度约为 50 PetaFLOPS，而 Google 现在拥有超过 80 PetaFLOPS 的领域专用超级计算机。

- 提示工程(prompt engineering)

高度训练的 Transformer 可以根据以自然语言输入的提示来执行任务。虽然提示是以自然语言输入的，但所用的词汇还是需要一定的结构，从而使提示成为一种元语言。

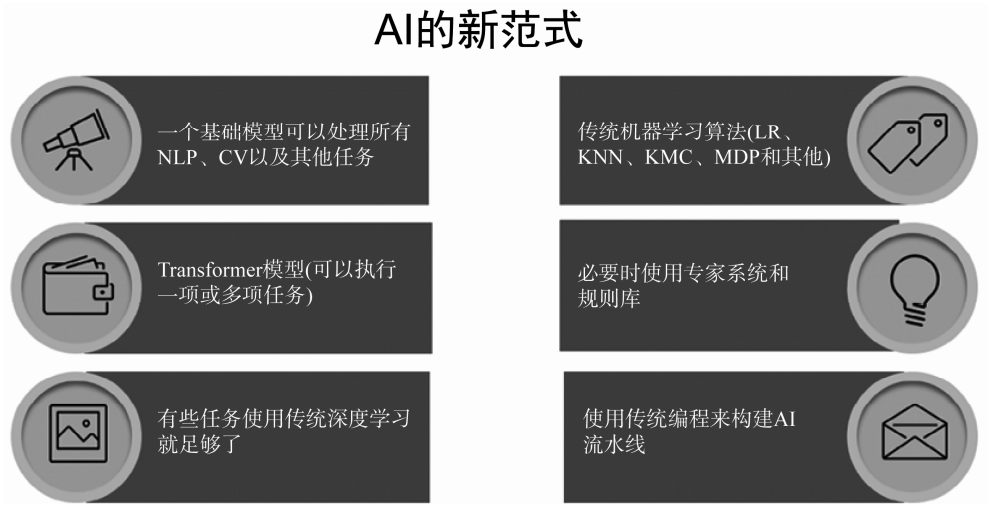

图 1.2 工业 4.0 AI 专家的技能要求

并非所有 Transformer 模型都可以称为基础模型。基础模型是指在超级计算机上用数十亿个参数对数十亿条数据进行训练得出的 Transformer 模型。这种模型不需要进一步微调即可执行各种任务。我们可以看到，基础模型的规模是前所未有的。这些经过大量训练的模型通常称为引擎。按照这个定义，只有 GPT-3、Google BERT 和少数 Transformer 引擎有资格称为基础模型。

本书后面提到的基础模型一般指 OpenAI GPT-3 或 Google BERT 模型。这是因为 GPT-3 和 Google BERT 在超级计算机上进行了大量训练。其他模型由于算力和数据不足，虽然有趣且有效，但用途有限，其同质化水平未能满足基础模型的定义。

接下来列举一个示例，演示基础模型如何改变我们的编程方式。

编程将成为 NLP 的一个子领域？

Chen et al. (2021)于 2021 年 8 月发表了一篇关于 Codex 的重磅论文。Codex 是一个可将自然语言转换为程序源代码的 GPT-3 模型。Codex 使用了 5400 万个公开的 GitHub

代码存储库进行训练。Codex 可以根据自然语言生成程序源代码，正如我们将在第 16 章所看到的那样。

这么一说，编程将变成从自然语言到程序源代码语言的翻译任务了？

编程将成为 GPT-3 引擎的一个 NLP 任务？

在回答这些问题之前，我们先看一个实例。

注意，Codex 是一种随机算法，因此对元语言要求很高。如果没有认真设计出正确的提示，则可能无法生成预期的内容。

我在试验 Codex 时创建了一些提示。注意，这个例子只是为了让人们了解 Codex 是如何工作的，纯粹出于教学目的，所以这些提示不一定适用于实际工作。

我的提示是：

- “generate a random distribution of 200 integers between 1 and 100” in Python(用 Python 生成 200 个整数(1～100)的随机分布)
- “plot the data using matplotlib”(使用 matplotlib 绘制数据)
- “create a k-means clustering model with 3 centroids and fit the model”(创建具有 3 个质心的 k 均值聚类模型并拟合模型)
- “print the cluster labels”(打印聚类标注)
- “plot the clusters”(绘制聚类图)
- “plot the clusters with centroids”(基于质心绘制聚类)

Codex 能自动将以上自然元语言提示翻译成 Python 代码！

注意，Codex 是一个随机模型，因此如果你按照以上自然语言试验，生成的代码可能与下文不同。

Codex 最终自动生成如下的 Python 代码，可供复制和测试：

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets.samples_generator import make_blobs

# generate random data
np.random.seed(0)
X, y = make_blobs(n_samples=200, centers=3, n_features=2, cluster_std=2,
random_state=0)

# plot the data
plt.scatter(X[:, 0], X[:, 1], s=50)
plt.show()

# create the k-means model
kmeans = KMeans(n_clusters=3, random_state=0)

# fit the model to the data
kmeans.fit(X)
```

```
# print the cluster labels
print(kmeans.labels_)

#plot the clusters
plt.scatter(X[:, 0], X[:, 1], c=kmeans.labels_, cmap='rainbow')
plt.show()

# plot the clusters with centroids
plt.scatter(X[:, 0], X[:, 1], c=kmeans.labels_, cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
c='black', s=100, alpha=0.5)
plt.show()
```

可以复制粘贴以上代码进行测试。你会发现以上代码是有效的。你也可以试试用 Codex 生成 JavaScript 代码。

GitHub Copilot 现在可以与一些 Microsoft 开发工具一起使用，我们将在第 16 章讲述更多。如果你学习了提示工程元语言，你将能够在未来几年内缩短开发时间。

未来随着 GitHub Copilot 的发展，很可能最终用户只需要掌握提示工程元语言，就可以创建原型或完成编程小任务。

我们将在第 16 章看到 Codex 在 AI 未来中的位置。

现在聊聊 AI 专家的光明未来。

AI 专家的未来

基础模型的社会影响不容小觑。提示工程已成为 AI 专家所需的技能。然而，AI 专家的未来不仅仅限于 Transformer 模型。AI 和数据科学在工业 4.0 中会有重叠。

AI 专家还是需要使用传统 AI、物联网、边缘计算等技术的机器-机器算法，还是需要使用传统算法来设计和开发机器人、服务器及各种设备之间的连接。

因此，本书不仅限于快速工程，还包括成为“工业 4.0 AI 专家”所需的各种设计技能。

提示工程是 AI 专家必须学习的设计技能的一个子集。在本书中，我将未来的 AI 专家称为“工业 4.0 AI 专家”。

现在让我们大致了解一下 Transformer 如何优化 NLP 模型。

1.2 使用 Transformer 优化 NLP 模型

几十年来，循环神经网络(RNN)和 LSTM 一直将神经网络应用于 NLP 序列模型。然而，当面对长序列和大量参数时，循环神经网络因为其局限性而无法进行很好的处理。从而导致目前最先进的 Transformer 模型占据主导地位。

本节将简要介绍 NLP 的背景，从而引出 Transformer 模型，我们将在第 2 章对其进行更详细的描述。这里我们先直观地了解一下 Transformer 模型中注意力头(attention

head)的工作原理，它在 NLP 中取代了 RNN。

Transformer 模型的核心概念可以简单地概括为混合词元。NLP 模型首先将词序列转换为词元(token)。RNN 通过循环函数分析词元。而 Transformer 模型不是按顺序分析词元，而是将每个词元与序列中的其他词元相关联，如图 1.3 所示。

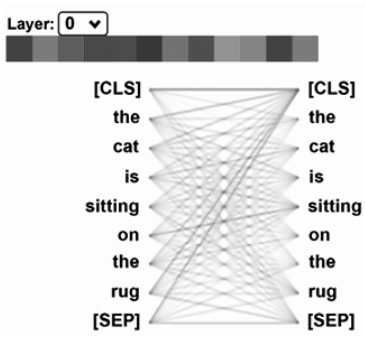

图 1.3　Transformer 模型中一层的注意力头

我们将在第 2 章详细介绍注意力头。目前，图 1.3 的要点是序列中的每个单词(词元)与序列中的所有其他单词相关。这种模型为工业 4.0 的 NLP 打开了大门。

这里简要介绍一下 Transformer 的背景。

Transformer 的背景

在过去 100 多年里，许多伟大的思想家致力于序列模式和语言建模。基于这些，机器逐渐学会了如何预测可能的词序列。这些伟人的壮举需要一整本书才能列举完。

因此在这个简短的一节里，我只能与你分享一些我最喜欢的研究者，他们为 Transformer 的到来奠定了基础。

20 世纪初，安德烈·马尔可夫(Andrey Markov)引入了随机值的概念，并创建了随机过程的理论。在 AI 中，我们将其称为马尔可夫决策过程(MDP)、马尔可夫链和马尔可夫过程。马尔可夫表明，只使用过去的元素就可以预测链、序列的下一个元素。他将这种方法应用于一个包含数千个字母的数据集，利用过去的序列来预测句子的下一个字母。记住，当时还没有计算机，但他提出了这个今天仍在 AI 领域中使用的理论。

1948 年，克劳德·香农(Claude Shannon)的《通信的数学理论》出版。克劳德·香农为基于源编码器、发射器和接收器或语义解码器的通信模型奠定了基础。他创造了我们今天所知的信息论。

1950 年，艾伦·图灵(Alan Turing)发表了他的重要文章《计算机与智能》。艾伦·图灵在这篇文章中基于二战期间成功解密德国消息的图灵机，对机器智能进行了讲述。这些德国消息由一系列单词和数字组成。

在 1954 年，乔治敦-IBM(Georgetown-IBM)实验室使用计算机通过一个规则系统将俄语句子翻译成英语。规则系统是一个运行一系列规则、用于分析语言结构的程序。即使在今天，规则系统依旧经常使用。然而在某些情况下，机器智能可通过自动学习模式来取代规则列表，以处理数十亿种语言组合。

AI 这个词最早是由约翰·麦卡锡(John McCarthy)在 1956 年提出的，从那时起，人们确定了机器是可以学习的。

1982 年，约翰·霍普菲尔德(John Hopfield)提出一种称为霍普菲尔德网络或关联

神经网络的 RNN。约翰·霍普菲尔德受到 W.A. Little 的启发，后者在 1974 年写了《大脑中持久状态的存在》一书，为持续至今几十年的学习过程奠定了理论基础。RNN 不断发展，最终发展为我们今天所熟知的形式 LSTM。

RNN 可以有效地记忆序列的持久状态，如图 1.4 所示。

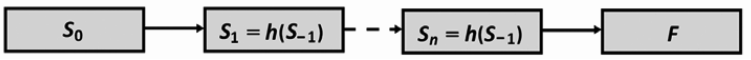

图 1.4 RNN 过程

每个状态 S_n 捕获 S_{n-1} 的信息。网络末端的函数 F 将执行一个操作：转导、建模或任何其他类型的基于序列的任务。

到了 20 世纪 80 年代，Yann LeCun 设计了多功能的卷积神经网络(CNN)。他将 CNN 应用于文本序列(也适用于序列转换和建模)。CNN 还基于 W.A. Little 提出的持久状态，逐层处理信息。然后到了 20 世纪 90 年代，经过数年的工作总结，Yann LeCun 制作出 LeNet-5，这催生了我们今天所知道的许多 CNN 模型。然而，在处理冗长而复杂的序列中的长期依赖关系时，CNN 原本高效的架构遇到瓶颈，达到了极限。

为突破 CNN 的极限，注意力的概念诞生了：注意力不仅关注序列中的最后一个词元，还关注其他词元。人们开始将注意力添加到 RNN 和 CNN 模型中。

之后，如果需要分析更长的序列则需要增加计算机算力，AI 开发者将使用更强大的机器，并找到优化梯度的方法。

研究人员对序列-序列模型继续进行研究，但结果并未达到预期。

似乎没有其他办法可以取得更多进展。就这样过了 30 年。然后到了 2017 年，工业化的最先进的 Transformer 出现了，它带来了注意力头子层等更多功能。RNN 不再是序列建模的先决条件了。

在深入研究原始 Transformer 的架构(我们将在第 2 章介绍)之前，先从高层面开始介绍我们应该使用的软件资源的范式变化。

1.3 我们应该使用哪些资源

工业 4.0 AI 模糊了云平台、框架、库、语言和模型之间的界限。Transformer 虽然是一门新技术，但是生态系统的范围之广和数量之多却令人惊叹。Google Cloud 提供了可直接使用的 Transformer 模型。

OpenAI 提供了一个几乎不需要编程的 Transformer API。Hugging Face 提供了模型云服务，而且模型数量还十分多。

本章将本书中将要实现的一些 Transformer 生态系统进行高层次的分析。

在 NLP 中，选择使用哪种资源来实现 Transformer 模型非常重要。这关系到项目的生存问题。想象一下在现实生活中的面试或演示。想象一下你正在与未来的雇主、

你当前的雇主、你的团队或客户交谈。

例如，如果你只会使用 Hugging Face。在面试时，你可能遇到这样一个负面反馈：很抱歉，我们在这类项目中使用的是 Google Trax，而不是 Hugging Face。你会使用 Google Trax 吗？如果你不会，那么面试可能会失败。

同样的问题也可能出现在专门使用 Google Trax 的情况下。你只会 Google Trax，但可能遇到一个希望使用 OpenAI GPT-3 引擎和 API 而不需要编程的面试官。如果你专门使用 OpenAI GPT-3 引擎和 API 而不会编程，你可能遇到一个更喜欢 Hugging Face 的 AutoML API 的项目经理或客户。最糟糕的情况是，对方接受了你的解决方案，但实际上你的方案并不适用于该项目的 NLP 任务。

要记住的关键概念是，如果你只关注你喜欢的解决方案，很可能在某个时刻会和船一起沉没。

你应该随着市场变化和项目需要而选择和学习适合的解决方案，而不是根据你个人喜好去选择解决方案。

本书并不旨在解释市场上存在的每个 Transformer 解决方案。相反，本书旨在详细解释 Transformer 生态系统，让你能够灵活适应在 NLP 项目中遇到的任何情况。

本节将介绍你将面临的一些挑战。我们先从 API 开始。

1.3.1　Transformer 4.0 无缝 API 的崛起

我们现在已经进入 AI 的工业化时代。Microsoft、Google、Amazon AWS 和 IBM 等公司提供了任何个人或中小企业都无法超越的 AI 服务。在训练 Transformer 模型和其他 AI 模型方面，科技巨头拥有价值百万美元的超级计算机和海量数据集。

大型科技巨头拥有广泛的企业客户群体，这些客户已经在使用它们的云服务。因此，将 Transformer API 添加到现有的云架构中所需的工作量比其他解决方案要少。

一个小公司甚至个人都可通过 API 访问最强大的 Transformer 模型，几乎不需要在开发上投入任何资金。一个实习生可在几天内实现 API。对于这样简单的实现，不需要成为工程师或拥有博士学位就能启动 AI 项目。

例如，OpenAI 平台现在提供了一种基于 SaaS(软件即服务)的 API，可以使用市场上一些最有效的 Transformer 模型。

OpenAI 的 Transformer 模型非常高效和接近人类，以至于当前的政策要求潜在用户填写申请表才能使用。不过当申请通过后，用户就可访问 NLP 的世界了！

OpenAI 的 API 的简洁性让用户感到惊讶：

(1) 一键获取 API 密钥

(2) 导入 OpenAI

(3) 通过提示输入你想要处理的任何 NLP 任务

(4) 你将收到由一段词元组成的回答结果

就这么简单！欢迎来到第四次工业革命和 AI 4.0 时代！

专注于纯代码解决方案的工业 3.0 程序员将演变为具有跨学科思维的工业 4.0 程序员。

工业 4.0 程序员将学习如何设计方法来向 Transformer 模型展示我们期望的内容，而不是像 3.0 开发者那样直观地告诉它该做什么。我们将在第 7 章通过 GPT-2 和 GPT-3 模型来探索这种新方法。

AllenNLP 为 Transformer 提供了免费使用的在线教学界面。AllenNLP 还提供了一个可以安装在 Jupyter 笔记本中的库。例如，假设要实现指代消解(Coreference resolution)。我们可从在线运行一个示例开始。

指代消解任务是指找到一个代词具体所指的实体，我们以图 1.5 的句子为例进行说明。

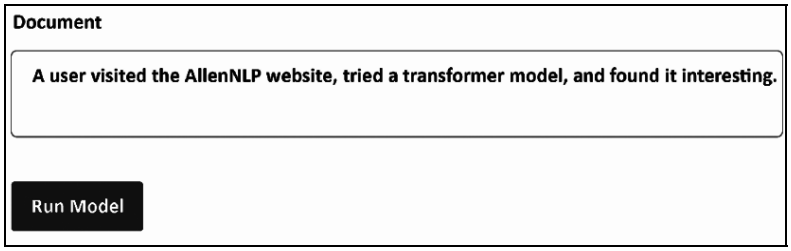

图 1.5 在线运行 NLP 任务

it 这个词可以指代 website 或 transformer model。在本例中，类 BERT 模型决定将 it 链接到 transformer model。单击 Run Model 按钮后，AllenNLP 将给出如图 1.6 所示的格式化输出。

图 1.6 AllenNLP Transformer 模型的输出

可以在 https://demo.allennlp.org/coreference-resolution 运行以上示例。注意，Transformer 模型会不断更新，所以你可能得到不同的结果。

尽管 API 可以满足许多需求，但它们也有局限性。一个多功能的 API 可能在所有任务中表现得相当不错，但对于特定 NLP 任务来说可能不够好。例如使用 Transformer 进行翻译表现经常不好。这种情况下，一个工业 4.0 的程序员、顾问或项目经理将不得不往更底层的方向走，例如使用即用型 API 驱动库。

1.3.2　选择即用型 API 驱动库

除了上节提到的 OpenAI 之外，还有很多即用型 API 驱动库，如 Google Trax。Google Trax 的使用也很简单，只需要几行代码即可在 Google Colab 中安装和运行。可以选择免费或付费的 Google Colab 服务。我们可以获取源代码，微调模型，甚至在自己的服务器或 Google 云上进行训练。因此，这些即用型 API 驱动库令使用现成的 API 定制一个用于翻译任务的 Transformer 模型变得可行。

我们将在第 6 章详细讲解 Google 在翻译方面的最新发展，并应用 Google Trax。

像 OpenAI 这样的 API 只需要很少的代码，而像 Google Trax 这样的库所需要的代码量会更多一些。这两种方法都表明，AI 4.0 API 需要在 API 的编辑器端编写更多代码，但在实现 Transformer 时所需要的工作量很少。

使用 Transformer 的最著名的在线应用程序之一是 Google 翻译。我们可以在线或通过 API 使用 Google 翻译。

让我们尝试使用 Google 翻译将一个需要指代消解的句子从英语翻译成法语。

Google 翻译似乎解决了指代消解的问题，但法语中的“transformateur”一词是指电子设备。因为 Transformer 这个词是法语中的新词。可见一个 AI 专家可能需要具备特定项目的语言和语言学技能。这种情况下，不需要太多的代码量。但该项目可能需要在请求翻译之前对输入进行清理(这点可能需要写很多代码)。

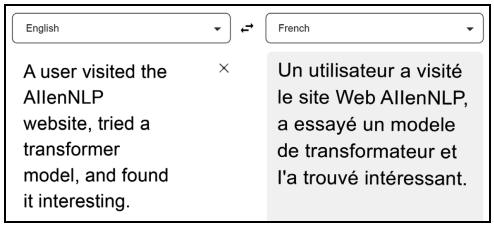

图 1.7　对指代消解示例使用 Google 翻译

这个例子表明，你可能需要与语言学家合作或者获得语言学技能来处理输入上下文。此外，在对输入进行清理的环节，可能需要进行大量的开发工作。

此外，我们需要写一点代码来调用 Google 翻译。或者还可能需要找到一个能够满足特定翻译需求的 Transformer 模型，如 BERT、T5 或其他将在本书中介绍的模型。

然而选择一个模型并非一件容易的事情。

1.3.3　选择 Transformer 模型

大型科技公司主导着 NLP 市场。仅 Google、Facebook 和 Microsoft 每天就运行着数十亿个 NLP 例程，这些例程反过来又提升了它们的 AI 模型，使模型有了无与伦比的能力。这些巨头现在对外提供各种 Transformer 模型(而且是排名靠前的基础模型)。

然而，一些较小的公司也进入了庞大的 NLP 市场。Hugging Face 现在也提供免费或付费的服务。对于 Hugging Face 来说，要达到 Google 研究实验室和 Microsoft 对 OpenAI 数十亿美元的资助所获得的性能将是一个挑战。基础模型的起点是 GPT-3 或 Google BERT 等在超级计算机上经过充分训练的 Transformer。

Hugging Face 选择了不同的方向，为任务提供了广泛和海量的 Transformer 模型，从而提供了灵活性；这种理念很有趣。除了提供了灵活的模型之外，Hugging Face 还提供了高级和开发者可控的 API。本书将用几章的篇幅详细讲述 Hugging Face。

除此之外，还有 OpenAI。OpenAI 专注于全球最强大的几个 Transformer 引擎，它们可以在许多 NLP 任务上达到人类水平。我们将在第 7 章展示 OpenAI GPT-3 引擎的强大能力。

以上这些策略经常相互冲突，但给我们提供了各种实施选项。最后我们必须介绍一下工业 4.0 AI 专家角色的技能要求。

1.3.4 工业 4.0 AI 专家的技能要求

工业 4.0 万物互联。机器直接与其他机器进行通信。由 AI 驱动的物联网信号触发自动化决策，不需要人为干预。NLP 算法发送自动化报告、摘要、电子邮件、广告等。

AI 专家将不得不适应这个越来越自动化的新时代，包括 Transformer 模型的实现。AI 专家的技能需要重新定义。如果我们按照从上到下的顺序列出一个 AI 专家需要完成的 Transformer NLP 任务，会发现一些高级任务对于 AI 专家来说几乎不需要写代码。一个 AI 专家可从以往繁重的细节任务脱身而出成为一个 AI 大师，从高层面提供设计思路、解释和实现。

对于 Transformer 的具体定义因生态系统而异。

让我们来看几个例子。

- API：使用 OpenAI API 来实现 AI 系统不需要 AI 程序员。只需要一个网页设计师设计和创建表单，然后由语言学家或领域专家写好提示，最后输入文本。在这种系统中，AI 专家需要具备语言技能，以展示和告诉 GPT-3 引擎具体如何完成任务，例如如何定义输入的上下文。这种新任务被称为提示工程(prompt engineering)。提示工程师在 AI 领域有很光明的前途！
- 库：使用 Google Trax 库来实现 AI 系统需要一定的编程工作，即如何使用现成的模型。AI 专家除了要精通语言学和 NLP 任务之外，还需要处理数据集和输出。
- 训练和微调：使用 Hugging Face 来实现 AI 系统需要完成一些编程工作，即调用 API 和库。这种情况下，训练、微调模型和找到正确的超参数将需要 AI 专家的专业知识。
- 开发级技能：在一些项目中，如第 9 章所述，词元分析器和数据集可能不匹配。这种情况下，能够与语言学家合作的 AI 程序员可以发挥关键作用。

最后值得一提的是，NLP AI 的最新发展——AI 助理正在颠覆 AI 开发生态系统。

- GPT-3 Transformer 目前嵌入了多个 Microsoft Azure 应用程序，如 GitHub

Copilot。正如本章 1.1.2 节“基础模型”介绍的那样，Codex 是我们将在第 16 章研究的另一个例子。

- 我们不能直接访问嵌入式 Transformer 模型；该模型可以提供自动开发支持，如自动生成代码。
- AI 助理对于终端用户来说是无感透明的，在辅助文本补全方面是无缝的。

要想直接访问 GPT-3 引擎，你首先需要创建一个 OpenAI 账户。然后可以使用 API 或直接在 OpenAI 用户界面中运行示例。

我们将在第 16 章探索 AI 助理。

综上所述，一个工业 4.0AI 专家的技能要求包括灵活性、跨学科知识。本书将为 AI 专家提供各种 Transformer 生态系统，以适应市场的新范式。

现在是时候总结本章了，第 2 章将深入讲述原始 Transformer 的迷人架构。

1.4　本章小结

第四次工业革命(或称工业 4.0)迫使 AI 进行更深的演化。第三次工业革命是数字化革命。而第四次工业革命则建立在数字革命的基础上，将万物互联。自动化流程正在取代人类在包括 NLP 在内的关键领域中的决策。

RNN 存在一些局限性，这些局限性在快节奏的现代世界中阻碍了自动化 NLP 任务的进展。Transformer 解决了这一问题，从而在摘要、翻译和各种 NLP 任务上帮助企业应对工业 4.0 的挑战。

因此，工业 4.0(I4.0)催生了一个 AI 产业化的时代。平台、框架、语言和模型概念的演变对于工业 4.0 开发者来说是一个挑战。基础模型通过提供同质模型来弥合第三次工业革命和第四次工业革命之间的差距，这些模型不需要进一步训练或微调即可执行各种任务。

然后我们讲述了一些资源，包括像 AllenNLP 这样的网站提供了不需要安装的教学性质的 NLP 任务供你学习，同时提供了在自定义程序中实现 Transformer 模型的资源。OpenAI 提供了一个 API，只需要几行代码就可以运行强大的 GPT-3 引擎。Google Trax 提供了一个端到端的库，Hugging Face 提供了很多 Transformer 模型和实现。我们将在本书介绍这些生态系统。

工业 4.0 是一次很大的变革，与以往的 AI 相比，需要具备更广泛的技能集。例如，项目经理可以要求网页设计师创建一个界面，使其可以与 OpenAI 的 API 进行交互来实施 Transformer。或者，在需要时，项目经理还可以要求 AI 专家下载 Google Trax 或 Hugging Face，使用定制的 Transformer 模型来完整地开发一个项目。

工业 4.0 对程序员来说是一次改变游戏规则的机会，他们的技能要求将扩展并需要更多的设计而不仅是编程。此外，AI 助理能够帮助我们编程。这些新的技能集是一个挑战，但也开启了新的令人兴奋的视野。

下一章将开始介绍原始 Transformer 的架构。

1.5 练习题

1. 我们仍处于第三次工业革命中。(对|错)
2. 第四次工业革命将实现万物互联。(对|错)
3. 工业 4.0 程序员有时不需要进行 AI 开发。(对|错)
4. 工业 4.0 程序员可能不得不从头开始实施 Transformer。(对|错)
5. 没必要学习多个 Transformer 生态系统，只学习一个(例如 Hugging Face)就足够了。(对|错)
6. 即用型 Transformer API 可以满足所有需求。(对|错)
7. 公司会接受开发商最了解的 Transformer 生态系统。(对|错)
8. 云 Transformer 已成为主流。(对|错)
9. Transformer 项目可以在笔记本电脑上运行。(对|错)
10. 对工业 4.0 AI 专家的灵活性要求会更高。(对|错)

第 2 章

Transformer模型架构入门

语言是人类交流的精髓。如果没有形成语言的单词序列，文明将永远不会诞生。我们的日常生活依赖于 NLP 将语言数字化：搜索引擎、电子邮件、社交网络、帖子、推文、智能手机短信、翻译、网页、流媒体网站上的语音转文本、热线服务上的文本转语音以及其他日常功能。

第 1 章解释了由于 RNN 的局限性，Transformer 云 AI 接管了相当一部分设计和开发。因此工业 4.0 开发者需要了解原始 Transformer 的架构以及基于其上的多个 Transformer 生态系统。

2017 年 12 月，Google Brain 和 Google Research 发表了开创性的“Attention is All You Need”论文(Vaswani et al.)。Transformer 诞生了。Transformer 的性能优于以往最先进的 NLP 模型。Transformer 的训练速度比以往的架构更快，并获得了更高的评估结果。因此 Transformer 成了 NLP 的关键组成部分。

Transformer 的注意力头(attention head)思想完全抛弃了循环神经网络。本章中，我们将打开 Vaswani et al. 2017 论文描述的 Transformer 模型的发动机盖，讲解 Transformer 模型架构的主要组件。我们将探索迷人的注意力世界，并讲解 Transformer 的关键组件。

本章涵盖以下主题：

- Transformer 的架构
- Transformer 的自注意力机制
- 编码器堆叠和解码器堆叠
- 将输入输出嵌入
- 位置编码
- 自注意力
- 多头注意力层
- 掩码多头注意力层
- 残差连接
- 规范化
- 前馈神经网络
- 预测

接下来我们开始深入了解原始 Transformer 模型的架构。

2.1 Transformer 的崛起：注意力就是一切

Vaswani et al. (2017)发表了他们的开创性论文“Attention is All You Need”。该论文相关的工作是在 Google Research 和 Google Brain 进行的。在本书中，我将把“Attention is All You Need”论文中描述的模型称为“原始 Transformer 模型”。

附录 A 讲解了 Transformer 模型引入的一些新术语。

本节我们将讲解原始 Transformer 模型的架构。然后接下来的各节将详细讲解模型每个组件内部的内容。

原始 Transformer 模型使用了 6 层堆叠。第 l 层的输出是第 l+1 层的输入，直到做出预测的最终层。图 2.1 的左侧是一个 6 层的编码器堆叠，右侧是一个 6 层的解码器堆叠。

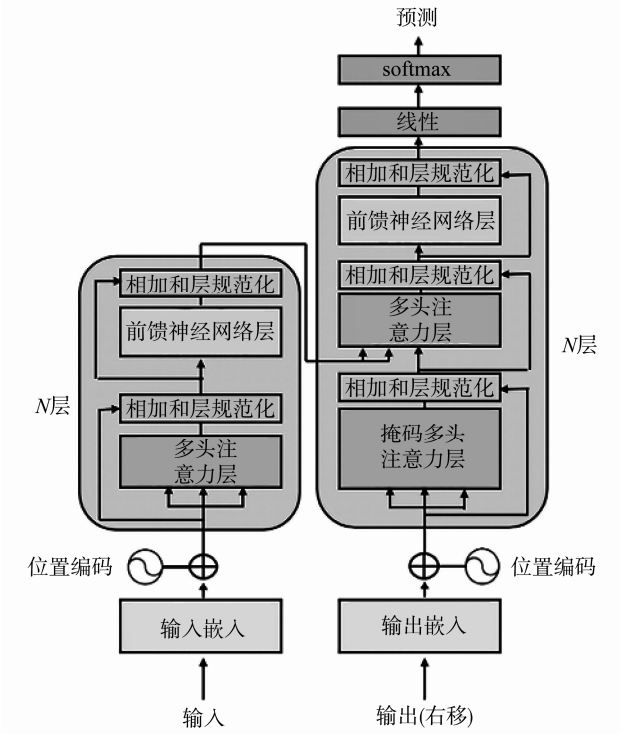

图 2.1 原始 Transformer 模型的架构