第 1 章

绪　论

自然语言处理(natural language processing,NLP)是研究如何利用计算机技术对语言文本(句子、篇章或话语等)进行处理和加工的一门学科,研究内容包括对词法、句法、语义和语用等信息的识别、分类、提取、转换和生成等各种处理方法和实现技术[①]。这一术语大约出现在 20 世纪 70 年代末期或 80 年代初期。概括地讲,自然语言处理包括对文本的分析、理解和转换、生成两大方向。近年来,随着计算机网络和移动通信技术的普及应用,这一技术在人类社会、经济和国家安全等各个领域都发挥了越来越重要的作用,其处理数据也从单一的文本模态扩展到了与语音、图像和视频等多模态数据的融合。

在学习自然语言处理技术的时候,我们不仅需要了解其中的理论方法和模型,更需要掌握每一种方法的实现技巧和使用方法。为此,本书筛选出一批近年来较为经典的模型和算法,从技术实现的角度对其进行剖析,给出了程序代码,以供读者进行实践和练习。

本章首先对自然语言处理方法给予简要的回顾和概述,然后介绍书中程序代码实现的平台和使用方法,最后对全书的内容和组织结构进行简要说明。

1.1　自然语言处理方法概述

在 20 世纪 80 年代中期之前,自然语言处理主要采用基于模板和规则的方法,所依据的理论基础是以乔姆斯基(N. Chomsky)为代表的语言学家提出的句法结构理论等。从 1947 年维沃(W. Weaver)和布斯(A. Booth)最早提出机器翻译概念到 80 年代中期,长达近 40 年的自然语言处理技术发展阶段通常被认为是基于符号逻辑推理的理性主义时期。

20 世纪 80 年代中期,基于语料库的统计自然语言处理方法逐渐萌生,n 元语法(n-gram)模型(或称 n 元文法模型)和隐马尔可夫模型(hidden Markov model,HMM)被引入自然语言处理领域,并得到广泛应用。日本学者长尾真(M. Nagao)提出了基于事例(实例)的机器翻译(example-based machine translation)方法。1989 年首个基于语料库的统计机器翻译系统诞生,随后一系列开源数据、工具、平台和系统的推广应用,将数据驱动的经验

① 见《计算机科学技术百科全书》(清华大学出版社,2018 年)第 1222 页。

主义方法提升到一个新的历史发展阶段，并持续了 20 余年。

统计自然语言处理方法可以大致划分为两类：生成式模型(generative model)和区分式模型(discriminative model)。生成式模型的基本思路是：假设 O 是观察值，Q 是模型，那么首先建立样本的概率密度模型 $P(O|Q)$，然后利用该模型进行推理预测。该方法建立在统计学和贝叶斯理论基础之上，要求已知样本无穷多或者尽量多。

区分式模型又称判别式模型，其基本思路是对条件概率(后验概率)$P(Q|O)$进行建模，在有限样本条件下建立判别函数，寻找不同类别之间的最优分类面。基于这种思路，很多自然语言处理任务转变为分类任务，或者序列标注任务。

当然，在某些任务上生成式模型和区分式模型可以集成应用。统计方法常用的模型归纳见表 1-1。

表 1-1 统计自然语言处理方法概览

类 别	代表模型	典型应用
生成式模型	n 元文法(n-gram)模型	汉语分词，机器翻译，自动写作
	隐马尔可夫模型(hidden Markov model，HMM)	汉语分词与词性标注，命名实体识别，语音识别
区分式模型	朴素贝叶斯分类器(naive Bayes，NB)	词义消歧，词性标注，各种分类任务
	决策树(decision tree)	词性标注，句法分析，各种分类任务
	k-近邻法(k-nearest neighbor，k-NN)	文本分类、聚类，情感分析
	最大熵(maximum entropy，ME)	词义消歧，句法分析，各种分类
	感知机(perceptron)	汉语分词与词性标注，各种分类任务
	支持向量机(support vector machine，SVM)	文本分类、情感分类等各种分类任务
	条件随机场(conditional random field，CRF)	汉语分词、命名实体识别等，各种分类或序列标注任务
混合模型	n 元文法模型＋条件随机场等	汉语分词与词性标注等各种序列标注任务

在统计方法建立的初期，几乎所有的自然语言处理任务都曾采用 n 元文法模型解决，从拼音-文字转换、汉语自动分词、命名实体识别等关键技术到机器翻译等应用系统，n 元文法模型几乎“无所不能”。之后，由于区分式方法不需要复杂的模型推导，增加或减少特征都容易实现，因此，区分式模型(通常表现为分类或序列标注模型)就像之前的 n 元文法模型或现在的 Transformer 模型，遍地开花。

无论如何，统计方法时期是自然语言处理技术发展历程中非常重要的一个历史阶段。尽管多数模型在几乎所有任务上的性能表现都已被神经网络方法超越，目前已经不再被广泛使用，但是很多建模思想，尤其是 n 元文法模型，对于后来神经语言模型的建立都有重要的借鉴和启发意义。

进入 21 世纪之后，在已有的前馈神经网络(feedforward neural network，FNN)、卷积神经网络(convolutional neural network，CNN)和循环神经网络(recurrent neural network，RNN)的基础上研发出了一系列新的模型，并被广泛应用于各类自然语言处理任务，逐渐形成了基于神经网络的自然语言处理方法。从早期的神经网络语言模型(neural network language model，NNLM)，到词嵌入(word embedding)方法、序列到序列(sequence-to-sequence，Seq2Seq)方法、注意力机制(attention mechanism)，再到 Transformer 和预训练

语言模型(pre-training language model)等一系列方法相继被提出,一再刷新各种自然语言处理任务的性能指标,各种工具和平台的开放使用极大地降低了自然语言处理领域的入门门槛,并为打通不同模态数据处理方法之间的壁垒奠定了基础,多模态信息融合处理成为可能。自然语言处理技术的应用也从早期以分析和理解为主、语言生成为辅,发展到分析、理解与生成齐头并进的新阶段。

纵览基于神经网络的自然语言处理方法,代表性的模型和工具归纳见表 1-2。

表 1-2 基于神经网络的代表性模型①

模 型	时 间	代表论文或网站	用途说明
长短时记忆模型(LSTM)	1997 年	(Hochreiter and Schmidhuber, 1997)	属于循环神经网络(RNN)的一种形式,能够缓解长序列训练的梯度消失和爆炸等问题
神经网络语言模型(NNLM)	2003 年	(Bengio et al., 2003)	学习单词的分布式表示,计算单词序列的概率
图神经网络(GNN)	2009 年	(Scarselli et al., 2009)	利用结构信息表征文本,在知识图谱、句法分析、语义角色标注和关系抽取等任务中广泛应用
词嵌入模型(word embedding): CBOW, skip-gram	2013 年	(Mikolov et al., 2013a,b)	生成词的分布式向量表示
门控循环单元(GRU)	2014 年	(Cho et al., 2014)	是 LSTM 的一种变体,简化了 LSTM 的模型结构,但能保持相当的性能
Seq2Seq	2014 年	(Sutskever et al., 2014)	序列到序列建模,输入和输出序列的长度是可变的
对抗生成网络(GAN)	2014 年	(Goodfellow et al., 2014)	用于对话系统、鲁棒机器翻译系统训练等任务
注意力机制(attention mechanism)	2015 年	(Bahdanau et al., 2015)	选择与当前目标最相关的信息进行处理
残差网络(ResNet)	2016 年	(He et al., 2016)	使用跳跃连接策略实现信号跨层传播,缓解了深层网络训练困难的问题
Transformer	2017 年	(Vaswani et al., 2017)	编码器-解码器转换模块,完全依靠自注意力机制计算输入和输出的表示,对长序列建模效果更好,且可以做到并行化计算
ELMo	2018 年	(Peters et al., 2018),又见网页②	预训练语言模型,采用基于 LSTM 的单向传统语言模型
GPT	2018 年 OpenAI	(Radford et al., 2018)	预训练语言模型,采用基于 Transformer 的单向传统语言模型
BERT	2018 年 Google	(Devlin et al., 2019)	预训练语言模型,采用基于双向自注意力机制的掩码语言模型

① 本表按模型提出的时间顺序排列。

② https://allenai.org/allennlp/software/elmo。

续表

模　型	时　间	代表论文或网站	用途说明
ERNIE	2019 年百度	(Sun et al., 2019)	预训练语言模型，采用融合实体知识的预训练语言模型
MASS	2019 年微软	(Song et al., 2019)	预训练语言模型，采用基于编码-解码框架的预训练语言模型
Pegasus	2020 年 Google	(Zhang et al., 2020)	预训练语言模型，采用编码-解码框架的预训练语言模型
GPT-3	2020 年 OpenAI	(Brown et al., 2020)	大规模预训练语言模型，采用 Transformer 解码器框架，参数达到 1750 亿
RocketQA	2021 年百度	(Qu et al., 2021)	开放领域问答语言模型
ERNIE-UIE	2022 年百度	(Lu et al., 2022)	预训练语言模型，采用知识增强的通用信息抽取统一框架的预训练语言模型
InstructGPT	2022 年 OpenAI	(Ouyang et al., 2022)	大规模预训练语言模型，采用基于指令学习和人类反馈学习的强化学习方法
LLaMA	2023 年 Meta	(Touvron et al., 2023)	大规模预训练语言模型，采用 Transformer 解码器框架，提供了 6.7 亿～65 亿不同参数规模的模型
GPT-4	2023 年 OpenAI	(OpenAI,2023)	大规模多模态预训练语言模型，能够接受文本和图像输入，并输出文本

随着神经网络模型的改进，以预训练语言模型为代表的各种模型表现出强大的功能，训练数据规模不断扩大，模型参数数量急剧膨胀，模型性能也在不断攀升。与此同时，模型可解释性和推理能力的提升得到了越来越多的关注。特别是 2022 年 OpenAI 公司的 ChatGPT 的发布，吸引了学术界和产业界对于大规模语言模型研究的广泛关注。实验表明，当模型参数规模超过一定数量时，预训练语言模型不仅能够显著提升多项自然语言处理任务的性能，而且会表现出多项特殊能力，如上下文学习(in-context learning)能力(Brown et al., 2020)、涌现能力(emergent abilities)(Wei et al., 2022a) 和思维链(chain of thought)能力等(Wei et al., 2022b)。为了凸显参数规模增大后的重要性，研究人员提出了“大语言模型(large language models，LLMs)或者预训练大模型”的概念，用以表示参数规模较大的语言模型。

尽管我们有理由相信，神经网络方法和基于神经网络的预训练模型不会是自然语言处理的终极方法，但是我们也不得不承认，这种方法已经成为本领域的主流，并将在未来很长一段时间内发挥主导作用，甚至成为解决某一类问题的垫脚石。

本书的主体内容将围绕神经网络方法展开，从神经网络基础模型和工具到关键技术和应用系统，通过不同的层次将当前主流的自然语言处理技术全方位地介绍给读者。本书提供的每行代码都由飞桨团队精心调试，通过动手实践，读者可以更好地理解自然语言处理技术的实现方法，并灵活地运用到各类任务上。

1.2 本书的内容组织

本书共包含15章内容，除了本章以外，其余14章将分别对n元文法模型、条件随机场及表1-1和表1-2中列出的部分经典模型的实现方法及其应用系统逐一介绍，按照从基础理论到关键技术，最终到应用系统的顺序依次展开。全书内容的组织关系如图1-1所示。

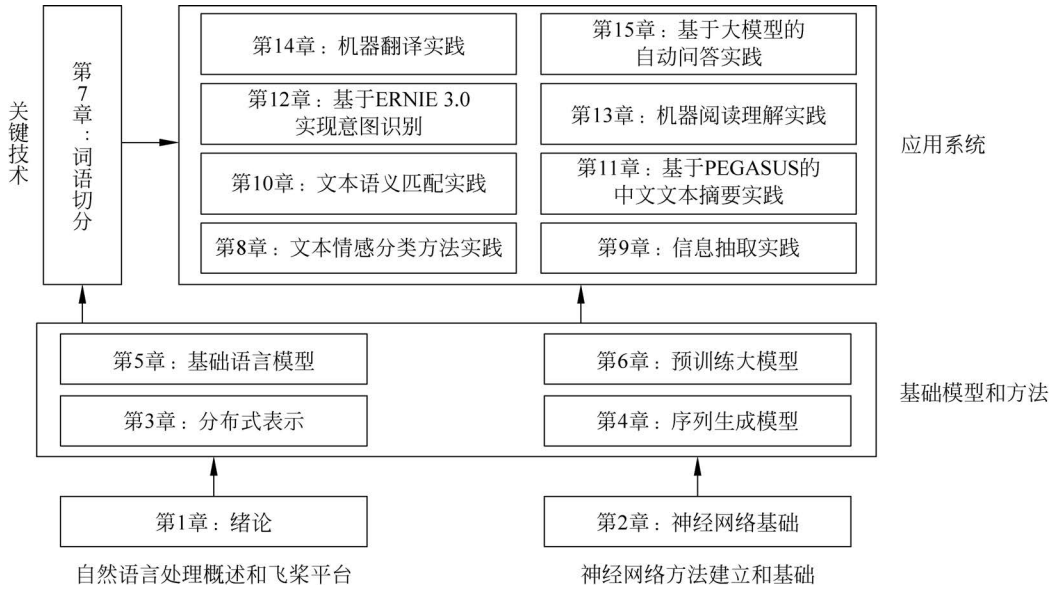

图1-1 本书的组织结构

第2章介绍神经网络基础模型的实现方法，包括前馈神经网络、卷积神经网络和传统的循环神经网络及其变种，如长短时记忆模型(long-short term memory，LSTM)、门控循环单元(gated recurrent unit，GRU)和双向的长短时记忆模型(bidirectional long-short term memory，BiLSTM)的实现方法。

第3章主要介绍两个最基本的词向量(word2vec)生成模型：连续的词袋(continuous bag-of-words，CBOW)模型和跳字(skip-gram)模型的实现方法，并以ELMo为例介绍动态词向量方法。以此为基础介绍短语和句子分布式表示的实现过程。

第4章介绍两个重要的序列生成模型的实现方法，包括序列到序列(Seq2Seq)建模和编码器-解码器转换模块(Transformer)的实现技术。

第5章介绍n元文法模型、基于前馈神经网络和LSTM的语言模型的实现方法。

第6章介绍3种预训练语言模型(GPT、BERT和ERNIE)的实现方法，并介绍预训练大模型的相关技术。

第2章～第6章介绍的内容可以认为是统计方法和神经网络方法的基础，后续所有的技术都是建立在这些模型和方法基础之上的。

第7章介绍自然语言处理，尤其是中文信息处理的关键技术——汉语词语自动切分。

第8章～第15章介绍自然语言处理应用系统的实现方法，包括情感分析、信息抽取、文本语义匹配、文本摘要、对话系统中的意图理解、机器阅读理解、机器翻译和基于大模型的问

答。需要说明的是，在这些章节中所介绍的系统实现方法只是基于某一种方法或模型完成，只是希望通过一个一个的具体事例介绍任务的实现过程，并非是性能达到最佳的实用系统。

本章只是对统计自然语言处理方法和基于神经网络的自然语言处理方法给予概括性的介绍，很多概念、模型和方法并未展开解释。对统计自然语言处理方法感兴趣的读者可以参阅文献（宗成庆，2013），想了解神经网络和深度学习方法的读者可以参阅文献（邱锡鹏，2020）。文献（宗成庆等，2022）和（Zong et al.，2021）对基于深度学习的自然语言处理方法有较为详细的介绍。文献（Zhang and Teng，2021）对统计方法和神经网络方法都有所涉及。后续各章将针对所介绍的内容给出相应的参考文献或网址，读者可以针对具体内容在实践中参考学习。

1.3 本书的实践平台

近年来，以深度学习为代表的人工智能技术得到了迅猛发展，神经网络结构越来越复杂，网络层级越来越深，参数规模也越来越大，尤其是大模型的兴起，使得从底层开始构建神经网络几乎是一件不可能完成的任务。同时大数据和大模型对模型训练的挑战也愈发严峻，需要具备超大规模分布式训练技术的深度学习框架来实现。

本书中的实践案例基于百度飞桨开发。飞桨（PaddlePaddle）是中国首个自主研发、功能丰富、开源开放的产业级深度学习平台，集核心框架、基础模型库、端到端开发套件、丰富的工具组件，以及 AI Studio 星河社区于一体。飞桨实现了动静统一的框架设计，兼顾灵活性和高效性；原生支持超大规模分布式训练能力，率先实现了多维混合并行策略和端到端自适应分布式训练技术，形成了显著的性能优势，支撑以文心一言为代表的文心大模型的高效训练和推理，效率和效果均获得大幅提升。

近年来，飞桨的生态建设取得显著进展，在科研院所、企事业单位、个人开发者群体中得到广泛应用。《深度学习平台发展报告（2022）》[①]指出，飞桨已经成为中国市场应用规模第一的深度学习框架和赋能平台。

实践环节提供两种运行方式：本地运行和 AI Studio 星河社区运行。下面分别介绍两种方式的操作方法，以及书中涉及的 API 和数据集。

1.3.1 本地运行

1. 环境准备

本地运行时，需要读者安装飞桨框架和 PaddleNLP。PaddleNLP 是基于飞桨框架开发的、简单易用且功能强大的自然语言处理开发库。在安装之前，需要先确认本机的运行环境，如操作系统、Python 和 pip 的版本是否满足飞桨的安装要求。飞桨支持安装在 Windows、macOS、Linux 等操作系统上，本节以 Windows 操作系统为例，介绍 pip 快速安装

① 见中国信通院《深度学习平台发展报告（2022 年）》。

的方法。

使用如下命令查看 Python 版本，目前飞桨支持的 Python 版本为 3.6/3.7/3.8/3.9/3.10。如果读者需要安装最新版本的 PaddleNLP，建议计算机的 Python 版本为 3.7 及以上。

```
python --version
```

使用如下命令查看 pip 版本，目前飞桨支持的 pip 版本为 20.2.2 或更高版本。

```
python -m ensurepip
python -m pip --version
```

使用如下命令确认 Python 和 pip 是 64bit 版本，且处理器架构是 x86_64（或称作 x64、Intel 64、AMD64）。

```
python -c "import
platform;print(platform.architecture()[0]);print(platform.machine())"
```

如果第一行输出的是“64bit”，第二行输出的是“x86_64”“x64”或“AMD64”，则满足飞桨的安装要求。

2. 快速安装

（1）安装飞桨框架

本书中的实践代码可以在 CPU 或 GPU 上运行，但除了 7.2 节和 8.1 节外，其余实践的模型结构较为复杂，需要更多的计算资源，在 GPU 上运行可以大大缩短模型训练的时间。如果读者本地没有 GPU 设备，建议选择在 AI Studio 星河社区上运行本书的实践代码，使用方法参见 1.3.2 节。

目前飞桨推荐的稳定版本是 2.5，建议读者安装最新的飞桨版本。使用如下命令安装 CPU 版本：

```
python -m pip install paddlepaddle==2.5.1 -i
https://pypi.tuna.tsinghua.edu.cn/simple
```

使用如下命令安装 GPU 版本：

```
python -m pip install paddlepaddle-gpu==2.5.1 -i
https://pypi.tuna.tsinghua.edu.cn/simple
```

使用如下命令验证安装的正确性：

使用 python 命令进入 Python 解释器，先输入 import paddle，再输入 paddle.utils.run_check()，如果出现“PaddlePaddle is installed successfully!”说明飞桨安装成功。

需要说明的是，目前飞桨 Windows 安装支持 CUDA 10.2/11.2/11.6/11.7 版本，如需使用其他 CUDA 版本，需要通过源码自行编译。最新的飞桨版本信息和其他环境的安装方法可以登录飞桨官网（https://www.paddlepaddle.org.cn）获取。

（2）安装 PaddleNLP

使用如下命令，采用 pip 的方式安装 PaddleNLP。

```
pip install --upgrade paddlenlp
```

1.3.2 AI Studio 星河社区运行

为了便于读者学习和实践，本书的实践案例均以 BML Codelab 的形式托管在 AI Studio 星河社区上。AI Studio 星河社区以飞桨和文心大模型为核心，集开放数据、开源算法、云端 GPU 算力及大模型开发工具于一体，为开发者提供模型与应用的高效开发环境。读者按照书中的说明可以直接使用 AI Studio 星河社区提供的免费计算资源在线编译运行书中的程序代码。

本书配套的实践项目和视频课程地址为：https://aistudio.baidu.com/course/introduce/28728。

AI Studio 星河社区已经默认安装最新版的飞桨，无需执行 1.3.1 节的安装操作。如图 1-2 所示，读者可以选择在基础版（CPU/DCU-16GB）、高级版（GPU-V100-16GB/GPU-V100-32GB）或尊享版（GPU-A100-40GB/GPU-4×V100-4×32GB）三种模式下运行项目，并通过每日运行项目、创建或 Fork 项目、优秀项目分享等方式获得免费的 GPU 算力。

图 1-2 项目运行环境

进入项目后，BML CodeLab 的页面布局如图 1-3 所示，由菜单栏、侧边栏、快捷工具栏和代码编辑区组成。本节简要介绍代码编辑区和侧边栏，更详细的使用方法可参见 BML

CodeLab 环境使用说明(https://ai.baidu.com/ai-doc/AISTUDIO/ Gktuwqf1x)。

图 1-3 BML CodeLab 的页面布局

图 1-3 的 BML CodeLab 项目页面代码编辑区由代码编写单元(Code Cell)和文本编辑单元(Mark Cell)组成。在代码编写单元内输入 Python 代码或 Linux 命令,单击“运行”按钮,代码或命令在云端执行,并将结果直接显示在项目页面中。在文本编辑区输入 Markdown 格式的文本(如文字、图片、数学公式等),方便编写项目说明。

侧边栏:在侧边栏单击“套件”,即可下载最新版本的 PaddleNLP,下载后的文件自动保存到/home/aistudio 中。

1.3.3 本书使用的 API

本书使用的 API 和功能介绍见表 1-3。

表 1-3 本书使用的 API 和功能介绍

任务环节	API 目录和示例		功能介绍
数据处理	Dataset		加载数据
	AutoTokenizer		通过指定模型名称,加载不同模型 Tokenizer 的接口
	Paddlenlp.data	DataCollatorWithPadding	将同批数据处理成相同的长度,方便模型处理
		DataCollatorForSeq2Seq DataCollatorForTokenClassification	NLP 任务相关的数据处理 API
		paddlenlp.data.Pad paddlenlp.data.Tuple paddlenlp.data.Stack	数据处理相关 API

续表

任务环节	API目录和示例		功能介绍
模型构建	AutoModel	AutoModel	通过AutoModel加载不同的预训练模型
		ErnieForQuestionAnswering AutoModelForCausalLM AutoModelForTokenClassification AutoModelForSequenceClassification	NLP任务相关的预训练模型API
	paddle.nn	paddle.nn.Transformer paddle.nn.lstm paddle.nn.GRU paddle.nn.Embedding	模型相关API，包括Transformer类预训练模型和循环神经网络模型
		paddle.nn.functional.dropout paddle.nn.functional.relu paddle.nn.functional.sigmoid paddle.nn.functional.label_smooth	激活函数相关API
模型训练	paddle.optimizer	paddle.optimizer.Adam paddle.optimizer.AdamW paddle.optimizer.SGD	优化算法相关API
	paddle.optimizer.lr	paddle.optimizer.lr.NoamDecay	学习率衰减相关API
模型配置	paddle.nn	paddle.nn.CrossEntropyLoss() paddle.nn.BCELoss() paddle.nn.MSELoss() paddle.nn.functional.cross_entropy paddle.nn.functional.mse_loss	损失函数相关API
模型评估	paddle.metric	paddle.metric.Accuracy() paddle.metric.Precision paddle.metric.Recall	评价指标相关API

1.3.4 本书使用的数据集

本书使用的数据集如下：

(1) icwb2数据集：在第7章使用，用于中文分词任务。icwb2数据集由台湾“中央研究院”、香港城市大学、北京大学和微软亚洲研究院共同建设，本书使用了北京大学提供的数据集，包含训练集18 056条(习惯上将“语句”表述为“条”)数据、验证集1 000条数据和测试集1 945条数据。

(2) IMDB数据集：在8.1节和8.2节使用，用于情感分析任务。IMDB是一份关于情感分析的二分类数据集，包含训练集和测试集各25 000条数据，每条数据都包含用户关于某一部电影的真实评价和对该电影的情感倾向。

(3) 百度自建的属性级情感分析数据集：在8.3节使用，用于属性级情感分析任务。包含训练集800条数据、验证集和测试集各100条数据，每条数据包括3项内容：原文本、属性文本和标签数据，其中属性文本由属性和观点词构成。