HZ BOOKS 华章IT

[PACKT] PUBLISHING

智能系统与技术丛书

Deep Learning for Natural Language Processing

基于深度学习的自然语言处理

[美] 卡蒂克·雷迪·博卡（Karthiek Reddy Bokka）
[印] 舒班吉·霍拉（Shubhangi Hora）
[德] 塔努吉·贾因（Tanuj Jain）
[美] 莫尼卡·瓦姆布吉（Monicah Wambugu）
著

赵鸣 曾小健 詹炜 译

机械工业出版社
China Machine Press

Deep Learning for Natural Language Processing

基于深度学习的自然语言处理

[美] 卡蒂克·雷迪·博卡（Karthiek Reddy Bokka）
[印] 舒班吉·霍拉（Shubhangi Hora）
[德] 塔努吉·贾因（Tanuj Jain）
[美] 莫尼卡·瓦姆布古（Monicah Wambugu）
著

赵鸣 曾小健 詹炜 译

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

基于深度学习的自然语言处理 /（美）卡蒂克·雷迪·博卡（Karthiek Reddy Bokka）等著；赵鸣，曾小健，詹炜译 . —北京：机械工业出版社，2020.5
（智能系统与技术丛书）
书名原文：Deep Learning for Natural Language Processing

ISBN 978-7-111-65357-8

I. 基… II. ① 卡 … ② 赵… ③ 曾… ④ 詹… III. 自然语言处理 IV. TP391

中国版本图书馆 CIP 数据核字（2020）第 062507 号

本书版权登记号：图字 01-2019-7265

基于深度学习的自然语言处理

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：李忠明 责任校对：李秋荣
印 刷：北京瑞德印刷有限公司 版 次：2020 年 5 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：14.75
书 号：ISBN 978-7-111-65357-8 定 价：79.00 元

客服电话：（010）88361066 88379833 68326294 投稿热线：（010）88379604
华章网站：www.hzbook.com 读者信箱：hzit@hzbook.com

THE TRANSLATOR'S WORDS

译者序

自然语言处理（Natural Language Processing，NLP）属于人工智能的一个子领域，是指用计算机对自然语言的形、音、义等信息进行处理，即对字、词、句、篇章的输入、输出、识别、分析、理解、生成等进行操作和加工。它对计算机和人类的交互方式有许多重要的影响。

本书可划分为三大部分：第一部分包括第 1、2 章，主要介绍了 NLP 的常用基本技术，包括词嵌入、文本规范化、标记文本、词性标注等，并且附有练习，以帮助读者实际上手和巩固所学知识；第二部分涵盖第 3 章到第 8 章，这部分专门针对用于 NLP 任务的神经网络与深度学习技术进行讲解，包括 CNN、RNN、GRU、LSTM 等，特别是第 8 章讲解了最前沿的用于自然语言处理任务的技术，包括注意力机制、transformer 及 BERT 等；第三部分（第 9 章）则是 NLP 在真正项目工作流中的体现。原理加项目代码实现是整本书的特点。希望读者可以多编码，加深记忆。

译者在本书翻译过程中参考了大量书籍和文献，但由于水平有限，译文中难免有不当之处，恳请读者批评指正。

曾小健

2020 年伊始

PREFACE

前　言

本书首先介绍自然语言处理领域的基本构件，接着介绍使用最先进的神经网络模型可以解决的问题，将深入涵盖文本处理任务中所需的必要预处理以及自然语言处理领域的一些热门话题，包括卷积神经网络、循环神经网络和长短期记忆网络。通过阅读本书，读者将理解文本预处理以及超参数调整的重要性。

学习目标

- ❑ 学习自然语言处理的基础知识。
- ❑ 了解深度学习问题的各种预处理技术。
- ❑ 使用 word2vec 和 GloVe 构建文本的矢量表示。
- ❑ 理解命名实体识别。
- ❑ 使用机器学习进行词性标注。
- ❑ 训练和部署可扩展的模型。
- ❑ 了解神经网络的几种架构。

目标读者

对自然语言处理领域的深度学习感兴趣的有抱负的数据科学家和工程师。

他们将从自然语言处理概念的基础开始，逐渐深入到神经网络的概念及其在文本处理问题中的应用。他们将学习不同的神经网络架构及其应用领域。需要具备丰富的 Python 知识和线性代数技能。

方法

本书从自然语言处理的基本概念讲起，在了解了基本概念之后，读者将逐渐意识到自然语言处理技术在现实世界中的应用和问题。接下来本书针对这些问题领域介绍开发解决方案的方法。本书还讨论了作为基于解决方案的方法的一部分的神经网络的基本构造块。最后通过实例阐述各种现代的神经网络架构及其相应的应用领域。

硬件要求

为了获得最佳体验，我们推荐以下硬件配置：

- 处理器：英特尔酷睿 i5 或同级产品
- 内存：4 GB 内存
- 存储：5 GB 可用空间

软件需求

我们还建议你预先安装以下软件：

- 操作系统：Windows 7 SP1 64 位、Windows 8.1 64 位或 Windows 10 64 位、Linux（Ubuntu、Debian、Red Hat 或 Suse）或 OS X 的最新版本。
- Python 3.6.5 或更高版本，最好是 3.7。可访问 https://www.python.org/downloads/release/python-371/ 下载。
- Jupyter（访问网站 https://jupyter.org/install 下载，按照说明安装）。或者，你可以使用 Anaconda 来安装 Jupyter。
- Keras（https://keras.io/#installation）。
- Google Colab 这是一个免费的 Jupyter 笔记本环境，运行在云基础架构上。强烈建议你使用它，因为其不需要任何设置，并且预先安装了流行的 Python 包和库（https://colab.research.google.com/note-books/welcome.ipynb）。

安装和设置

每一次伟大的旅程都是从一个不起眼的步骤开始的，对于即将到来的数据领域的冒险也不例外。在能够用数据做令人敬畏的事情之前，我们需要准备好最高效的环境。

在 Windows 上安装 Python

1）在官方安装页面（https://www.python.org/downloads/windows/）上找到你想要的

Python 版本。

2）确保根据你的计算机系统安装正确的“位”版本（32 位或 64 位）。你可以在操作系统的“系统属性”窗口中找到此信息。

下载安装程序后，只需双击文件，并按照屏幕上显示的用户友好提示操作。

在 Linux 上安装 Python

要在 Linux 上安装 Python，需执行以下操作：

1）在命令提示符下运行 **python3--version** 验证尚未安装 p\Python 3。

2）要安装 Python 3，请运行以下命令：

```
sudo apt-get update
sudo apt-get install python3.6
```

3）如果遇到问题，有许多在线资源可以帮助你解决问题。

在 macOS X 上安装 Python

要在 macOS X 上安装 Python，需执行以下操作：

1）通过按住“CMD + 空格”组合键打开终端，在打开的搜索框中键入终端，然后按回车键。

2）通过命令行运行 **xcode--select--install** 来安装 Xcode。

3）安装 Python 3 最简单的方法是使用 homebrew，通过命令行运行 **ruby--e"$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"** 来安装。

4）将 homebrew 添加到你的 PATH 环境变量中。通过运行 **sudo nano~/.profile** 在命令行中打开你的配置文件，并在底部插入 **export PATH="/usr/local/opt/python/libexec/bin:$PATH"**。

5）最后一步是安装 Python。在命令行中，运行 **brew install python**。

6）注意，如果你安装 Anaconda，最新版本的 Python 将自动安装。

安装 Keras

要安装 Keras，需执行以下步骤：

1）由于 **Keras** 需要另一个深度学习框架作为后端，你需要先下载另一个框架，建议使用 **TensorFlow**。

要在你的平台上安装 **TensorFlow**，请访问 https://www.tensorflow.org/install/。

2）安装后端后，就可以使用以下命令安装 **Keras**:

```
sudo pip install keras
```

也可以从 GitHub 安装它，使用以下方法克隆 **Keras**：

```
git clone https://github.com/keras-team/keras.git
```

3）使用以下命令在 Python 上安装 **Keras**：

```
cd keras
sudo python setup.py install
```

现在需要配置后端。更多信息请参考链接 https://keras.io/backend/。

下载示例代码及彩色图像

本书的示例代码及所有截图和样图，可以从 http://www.packtpub.com 通过个人账号下载，也可以访问华章图书官网 http://www.hzbook.com，通过注册并登录个人账号下载。

CONTENTS

目　录

CHAPTER 1

第 1 章

自然语言处理

学习目标

本章结束时，你将能够：

- 描述自然语言处理及其应用。
- 解释不同的文本预处理技术。
- 对文本语料库执行文本预处理。
- 解释 Word2Vec 和 GloVe 的词嵌入功能。
- 使用 Word2Vec 和 GloVe 生成词嵌入。
- 使用 NLTK、Gensim 和 Glove-Python 库用于文本预处理以及生成词嵌入。

本章旨在为你提供自然语言处理基础知识以及深度学习中使用的各种文本预处理技术。

1.1 本章概览

本书将指导你理解和优化深度学习技术，以进行自然语言处理，从而进一步推动强人工智能的实际应用。读者将了解自然语言处理的概念、应用和实现，并学习深度神经网络的方法，利用神经网络使机器理解自然语言。

1.2 自然语言处理的基础知识

为了便于理解，我们将这个术语分为两部分：

- 自然语言是一种有机且自然发展而来的书面和口头交流形式。
- 处理意味着使用计算机分析和理解输入数据。

如图 1-1 所示，自然语言处理是人类语言的机器处理，旨在教授机器如何处理和理解人类的语言，从而在人与机器之间建立一个简单的沟通渠道。

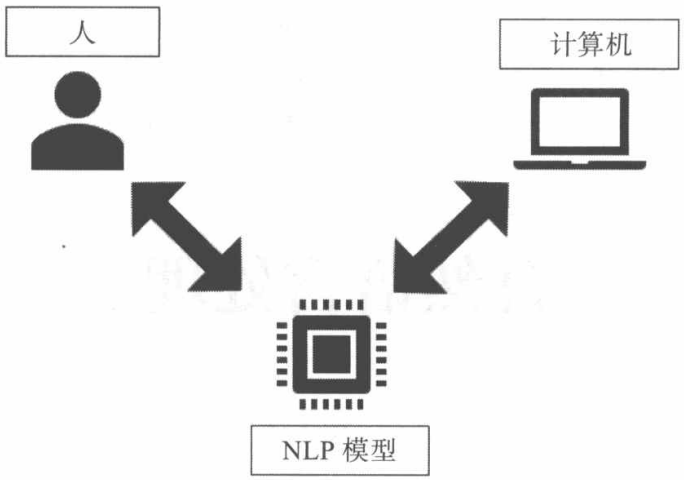

图 1-1　自然语言处理

自然语言处理的应用很广泛，例如，在我们的手机和智能音箱中的个人语音助手，如 Alexa 和 Siri。它们不仅能够理解我们的说话内容，而且能够根据我们说的话采取行动，并做出反馈。自然语言处理算法促进了这种与人类沟通的技术。

在上述自然语言处理定义中要考虑的关键是：沟通需要以人类的自然语言进行。几十年来，我们一直在与机器沟通：创建程序来执行某些任务并执行。然而，这些程序是用非自然语言编写的，因为它们不是口头交流的形式，也不是自然或有机发展而来的。这些语言，例如 Java、Python、C 和 C ++，都是在主要考虑机器的情况下创建的，并且始终考虑的是“机器能够轻松理解和处理的是什么？”

虽然 Python 是一种对用户更加友好的语言，且易于学习和编码，但与机器沟通，人类必须学习机器能够理解的语言。自然语言处理、机器学习、深度学习的关系如图 1-2 所示。

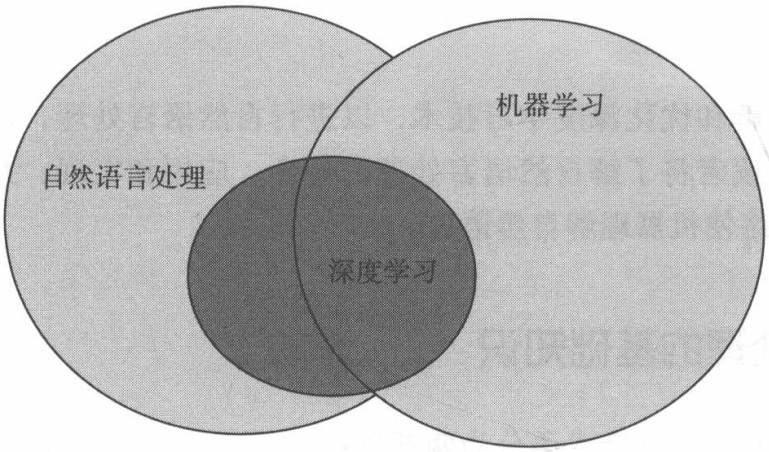

图 1-2　自然语言处理的维恩图

自然语言处理的目的与此相反。自然语言处理不是以人类顺应机器的方式学习如何有

效地与它们沟通，而是使机器能够与人类保持一致，并学习人类的交流方式。其意义更为重大，因为技术的目的本来就是让我们的生活更为轻松。

我们用一个例子来澄清这一点，你的第一个程序是一段让机器打印“hello world”代码。这是你顺应机器并要求它用其理解的语言执行任务。通过向其发出这个命令来要求你的语音助手说“hello world”，并做出“hello world”的反馈，就是自然语言处理应用的一个例子，因为你用自然语言与机器通信。机器符合你的沟通形式，理解你所说的内容，处理你要求它执行的操作，然后执行任务。

自然语言处理的重要性

图 1-3 说明了人工智能领域的各个部分。

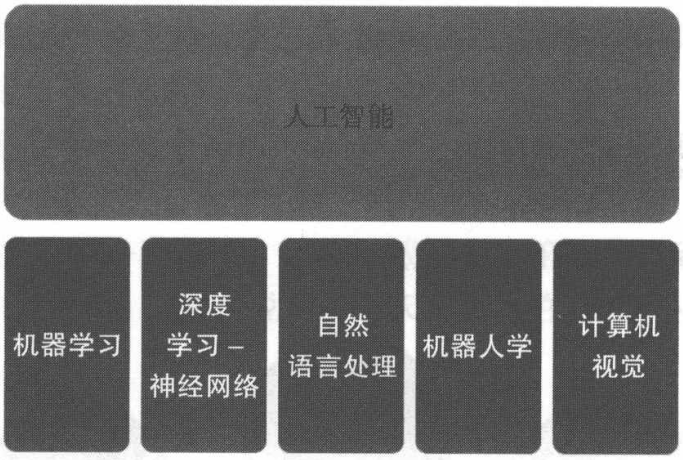

图 1-3 人工智能及其一些子领域

与机器学习和深度学习一样，自然语言处理是人工智能的一个分支，因为其处理自然语言，所以它实际上是人工智能和语言学的交叉。

如上所述，自然语言处理使机器能够理解人类的语言，从而在两者之间建立有效的沟通渠道。然而，自然语言处理的必要性还有另一个原因。那就是，像机器一样，机器学习模型和深度学习模型对数值数据最有效。数值数据对人类来说很难自然产生。很难想象我们用数字而不是语言交谈。因此，自然语言处理与文本数据一起工作，并将其转换成数值数据，从而使机器学习模型和深度学习模型能够适用于文本数据。因此，它的存在是为了通过从人类那里获取语言的口头和书面形式，并将它们转换成机器能够理解的数据，来弥合人类和机器之间的交流差距。得益于自然语言处理，机器能够理解并回答基于自然语言的问题、解决使用自然语言的问题以及用自然语言交流等。

1.3 自然语言处理的能力

自然语言处理有许多有益于人类生活的现实应用。这些应用程序属于自然语言处理的

三大功能：

❑ **语音识别**

机器能够识别自然语言的口语形式，并将其翻译成文本形式。比如智能手机上的听写，你可以启用听写功能并对着手机说话，它会将你所说的一切转换成文本。

❑ **自然语言理解**

机器能够理解自然语言的口语和书面语。如果给机器一个命令，它就能理解并执行。例如，在你的手机上对 Siri 说“嘿，Siri，打电话回家”，Siri 就会自动为你打电话回家。

❑ **自然语言生成**

机器能够自己生成自然语言。例如，在手机上对 Siri 说“ Siri，现在几点了？”Siri 回复说：“现在是下午 2:08”。

这三种能力用于完成和自动化许多任务。让我们来看看自然语言处理的一些应用。

注意　文本数据被称为语料库（corpora）或一个语料（corpus）。

1.4　自然语言处理中的应用

图 1-4 描述了自然语言处理的一般应用领域。

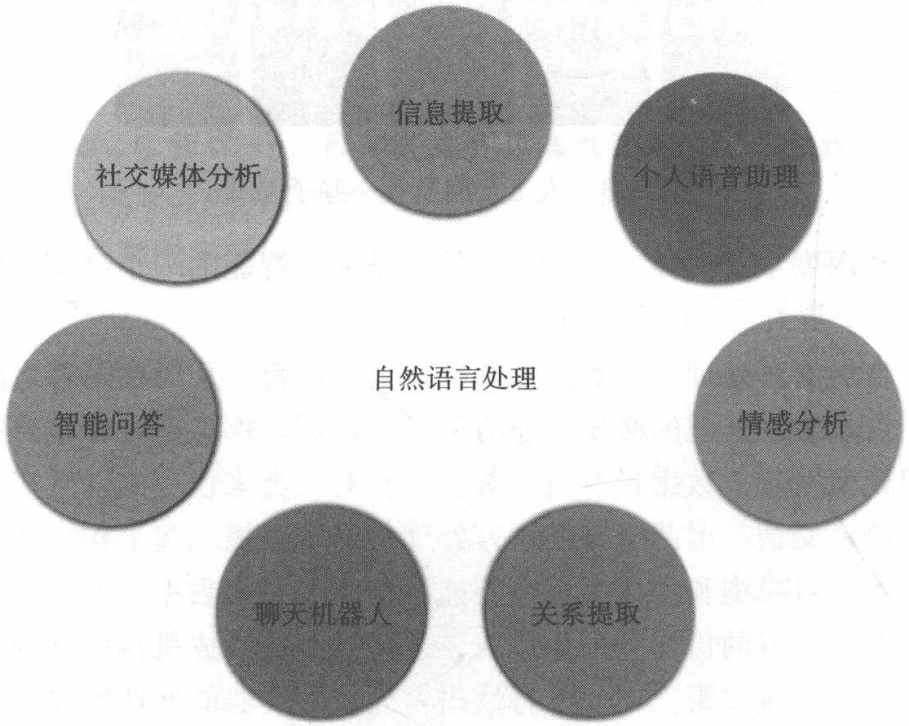

图 1-4　自然语言处理的应用领域

❑ **自动文摘**

包括对语料库生成摘要。

❑ 翻译

要求有翻译工具，以从不同的语言翻译文本，例如，谷歌翻译。

❑ **情感分析**

这也被称为情感的人工智能或意见挖掘，它是从书面和口头语料库中识别、提取和量化情感和情感状态的过程。情感分析工具用于处理诸如客户评论和社交媒体帖子之类的事情，以理解对特定事物的情绪反应和意见，比如新餐厅的菜品质量。

❑ **信息提取**

这是从语料库中识别并提取重要术语的过程，称为实体。命名实体识别属于这一类，将在下一章中解释。

❑ **关系提取**

关系提取包括从语料库中提取语义关系。语义关系发生在两个或多个实体（如人、组织和事物）之间属于许多语义类别之一。例如，如果一个关系提取工具被赋予了关于 Sundar Pichai 的内容，以及他是谷歌的 CEO，该工具将能够生成“ Sundar Pichai 就职于谷歌”作为输出，Sundar Pichai 和谷歌是两个实体，“就职于”是定义它们之间关系的语义类别。

❑ **聊天机器人**

聊天机器人是人工智能的一种形式，被设计成通过语音和文本与人类交流。它们中的大多数模仿人，使你觉得在和另一个人说话。聊天机器人在健康产业被用于帮助患有抑郁症和焦虑症的人。

❑ **社交媒体分析**

社交媒体的应用，如 Twitter 和 Facebook，都有标签和趋势，并使用自然语言处理来跟踪和监控这些标签和趋势，以了解世界各地正在交谈的话题。此外，自然语言通过过滤负面的、攻击性的和不恰当的评论和帖子来帮助优化过程。

❑ **个人语音助理**

Siri、Alexa、谷歌助手以及 Cortana 都是个人语音助理，充分利用自然语言处理技术来理解和回应我们。

❑ **语法检查**

语法检查软件会自动检查和纠正你的语法、标点和拼写错误。

1.4.1 文本预处理

在回答关于理解文章的问题时，由于问题针对文章的不同部分，因此一些词和句子对你很重要，有些则无关紧要。诀窍是从问题中找出关键词，并将其与文章匹配，以找到正确的答案。

文本预处理思想是这样的：机器不需要语料库中的无关部分。它只需要执行手头任务所需的重要单词和短语。因此，文本预处理技术涉及为机器学习模型和深度学习模型以及

适当的分析准备语料库。文本预处理基本上是告诉机器什么需要考虑、哪些可以忽略。

每个语料库根据需要来执行任务的不同文本预处理技术，一旦你学会了不同的预处理技术，你就会明白什么地方使用什么文本预处理技术和为什么使用。其中技术的解释顺序通常是被执行的顺序。

在下面的练习中，我们将使用 NLTK Python 库，但是在进行这些活动时可以随意使用不同的库。**NLTK** 代表**自然语言工具包**（Natural Language Toolkit），是自然语言处理最简单也是最受欢迎的 Python 库之一，这就是为什么我们用它来理解自然语言处理的基本概念。

注意　关于自然语言工具包的更多信息，请访问 https://www.nltk.org/。

1.4.2　文本预处理技术

以下是自然语言处理中最常用的文本预处理技术：

- ❑ 小写 / 大写转换
- ❑ 去噪
- ❑ 文本规范化
- ❑ 词干提取
- ❑ 词形还原
- ❑ 标记化
- ❑ 删除停止词

接下来分别介绍。

1. 小写 / 大写转换

这是人们经常忘记使用的最简单有效的预处理技术之一。它要么将所有的大写字符转换为小写字符，以便整个语料库都是小写的；要么将语料库中的所有小写字符转换为大写字符，以便整个语料库都是大写的。

当语料库不太大，并且任务涉及同一个词由于字符的大小写，而作为不同的术语或输出识别时，这种方法特别有用，因为机器固有地将大写字母和小写字母作为单独的实体来处理。比如，“A”与“a”是不同的。这种输入大小写的变化可能导致不正确的输出或根本没有输出。

例如，包含“India”和“india”的语料库如果不应用小写化，机器会把它们识别为两个独立的术语，而实际上它们都是同一个单词的不同形式，并且对应于同一个国家。小写化后，仅存在一种“India”实例，即“india”，简化了在语料库中找到所有提到印度时的任务。

注意　所有的练习和活动主要在 Jupyter Notebook 上开发。读者需要在系统上安装 Python 3.6 和 NLTK。

练习 1-6 可以在同一个 Jupyter notebook 上完成。