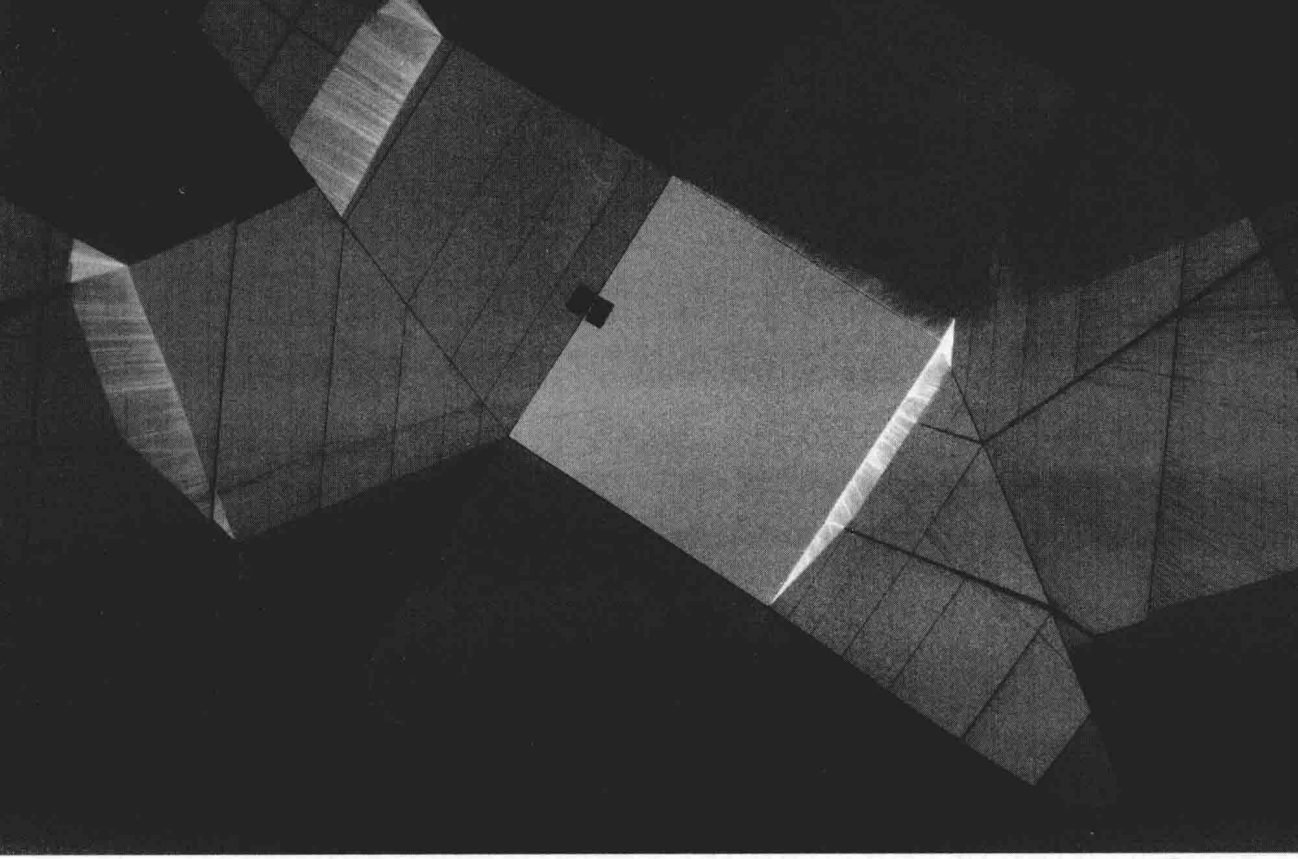
NLTK 基础教程

用NLTK和Python库构建机器学习应用

NLTK Essentials

[印度] Nitin Hardeniya 著
凌杰 译

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS



NLTK基础教程

用NLTK和Python库构建机器学习应用

[印度] Nitin Hardeniya 著

凌杰 译

人民邮电出版社

北京

图书在版编目（CIP）数据

NLTK基础教程 : 用NLTK和Python库构建机器学习应用 / (印) 哈登尼亚 (Nitin Hardeniya) 著 ; 凌杰译
. — 北京 : 人民邮电出版社, 2017.6
ISBN 978-7-115-45257-3

Ⅰ. ①N… Ⅱ. ①哈… ②凌… Ⅲ. ①软件工具－程序设计 Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2017)第079700号

版权声明

内容提要

NLTK 库是当前自然语言处理（NLP）领域最为流行、使用最为广泛的库之一，同时 Python 语言也已逐渐成为主流的编程语言之一。

本书主要介绍如何通过 NLTK 库与一些 Python 库的结合从而实现复杂的 NLP 任务和机器学习应用。全书共分为 10 章。第 1 章对 NLP 进行了简单介绍。第 2 章、第 3 章和第 4 章主要介绍一些通用的预处理技术、专属于 NLP 领域的预处理技术以及命名实体识别技术等。第 5 章之后的内容侧重于介绍如何构建一些 NLP 应用，涉及文本分类、数据科学和数据处理、社交媒体挖掘和大规模文本挖掘等方面。

本书适合 NLP 和机器学习领域的爱好者、对文本处理感兴趣的读者、想要快速学习 NLTK 的资深 Python 程序员以及机器学习领域的研究人员阅读。

◆ 著 [印度] Nitin Hardeniya
译 凌 杰
责任编辑 陈冀康
执行编辑 武晓燕
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：10.75
字数：210 千字 2017 年 6 月第 1 版
印数：1－3 000 册 2017 年 6 月北京第 1 次印刷

著作权合同登记号 图字：01-2015-8290 号

定价：49.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广字第 8052 号

作者简介

Nitin Hardeniya 数据科学家，拥有 4 年以上从业经验，期间分别任职于 Fidelity、Groupon 和[24]7 等公司，其业务横跨各个不同的领域。此外，他还拥有 IIIT-H 的计算语言学硕士学位，并且是 5 项客户体验专利的作者。

他热衷于研究语言处理及大型非结构化数据，至少拥有 5 年日常使用 Python 的工作经验。他相信，用 Python 可以构建出大部分与数据科学相关问题的单点解决方案。

他将自己写这本书的经历看成是自己职业生涯的众多荣誉之一，希望用一种非常简单的形式为人们介绍与 NLP 和机器学习相关的、所有的这些复杂工具。在这本书中，他为读者提供了一种变通方法，即使用一些相关特定能力的 Python 库，如 NLTK、scikit-learn、panda 和 NumPy 等。

审阅者简介

Afroz Hussain 数据科学家，目前在 PredictifyMe 公司从事与美国基础数据科学、机器学习起步相关的研究。他在数据科学领域拥有丰富的项目经验、多年使用 Python、scikit-learn，以及基于 NLTK 进行文本挖掘的工作经历。他拥有 10 年以上的编程经验以及与数据分析和商业智能项目相关的软件开发经验。此外，他还通过在线课程以及参加 Kaggle 比赛等活动，获得了不少数据科学领域的新技能。

Sujit Pal 目前就职于 Elsevier 实验室，这是一个包含了 Reed-Elsevier PLC 工作组在内的研发团队。他的兴趣主要集中在信息检索、分布式处理、本体开发、自然语言处理和机器学习这几个领域。而且，他也很喜欢用 Python、Scala 和 Java 来编写自己的代码。他充分整合了自己在这些方面的技能，帮助公司改进了不同产品的一些特性并构建了一些新特性。他深信自己需要终身学习，并且也在博客：sujitpal.blogspot.com 中分享其经验。

Kumar Raj 第二代数据科学家，目前就职于惠普软件的研发部门，为其提供相关的解决方案。在那里，他主要负责开发以惠普软件产品为核心的分析层。他毕业于印度理工学院 Kharagpur 技术分校，并具有两年以上各种大数据分析领域的工作经验，涉及文本分析、网页抓取及检索、人力资源分析、虚拟系统的性能优化，以及气候变化的预测等。

译者序

说来也凑巧，在我签下这本书的翻译合同时，这个世界好像还不知道 AlphaGo 的存在。而在我完成这本书的翻译之时，Master 已经对人类顶级高手连胜 60 局了。至少从媒体的热度来看，的确在近几年，人工智能似乎是越来越火了。其原因是 Google 在汽车驾驶和围棋这两个领域的项目得到了很好的进展和宣传，而这两个领域在过去被很多人想当然地认为是人类的专属领域。因此在专属领域接连被突破情况下，一些人得了“机器恐惧症”。例如高晓松先生的这段微博：

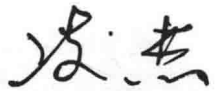
> @高晓松
> 作为自幼学棋，崇拜国手的业余棋手，看了 Master50：0 横扫中日韩顶尖高手的对局，难过极了。为所有的大国手伤心，路已经走完了。多少代大师上下求索，求道求术，全被破解。未来一个八岁少年只要一部手机就可以战胜九段，荣誉信仰灰飞烟灭。等有一天，机器做出了所有的音乐和诗歌，我们的路也会走完。
> 1 月 4 日 16:21 来自 iPhone 7 Plus

其实之所以会有这样恐惧，大部分是因为人们在讨论人工智能的时候容易将机器“人格化”，很多科幻作品就是这么干的，这看起来很合理，但问题是机器无论如何都不是人。对于机器来说，围棋说穿了不过是一种基于统计学概率的决策模型，属于数学领域的问题，它本来就是机器的强项。用围棋对于人类的难度来推导机器智能的进步，其实是很没有逻辑的事情。而且事实上，今天所流行的这些人工智能方法都是在 20 世纪 70 年代前后提出的理论，今天的辉煌主要是由于硬件的进步为实现提供了基础，但在智能上并没有多大的实质突破。要知道，人们对于鉴定人工智能的主要标准早有定论，那就是图灵测试。

图灵测试关注的是人机对话能力，换句话说，什么时候机器能通过对话骗到你的一百块钱，也比它下棋下赢世界冠军更智能点。而想要增强人机对话能力，自然语言处理就是

首当其冲的一个领域了。正如我们所说，机器的专长是数学领域，所以自然语言处理问题的目的就是要把我们人类的文本、音频转换成可被分析的数学模型，这对于机器来说是比围棋困难得多的事情。这也是人类和机器的根本区别，对于这两种智能来说，困难的定义是截然不同的。

说实话，刚开始译这本书的时候，我对它的翻译难度有些估计不足，很多专业词汇国内还似乎还没有标准译法。有些甚至根本找不到对应的中文翻译。虽然对于每个小节我都期望查阅大量的资料，尽量保证翻译的质量，但实在有点太累人了，太费时了，妥协、遗憾在所难免。在这里向读者们致歉，还希望你们多多包涵。同时也感谢人民邮电出版社的陈冀康编辑对于我拖稿行为的容忍，其实我还想再拖上半年的。



2017 年 1 月 10 日

于新安江畔

前言

这是一本介绍 NLTK 库，以及如何将该库与其他 Python 库搭配运用的书。NLTK 是当前自然语言处理（NLP）社区中最为流行、使用最为广泛的库之一。NLTK 的设计充分体现了简单的魅力。也就是说，对于大多数复杂的 NLP 任务，它都可以用寥寥几行代码来实现。

本书的前半部分从介绍 Python 和 NLP 开始。在这部分内容中，你将会学到一些通用的预处理技术，例如标识化处理（tokenization）、词干提取（stemming）、停用词（stop word）去除；一些专属于 NPL 领域的预处理技术等，如词性标注（part-of-speech tagging）；以及大多数文本相关的 NLP 任务都会涉及的命名实体识别（Named-entity recognition，简称 NER）等技术。然后，我们会逐步将焦点转到更为复杂的 NLP 任务上，例如语法解析（parsing）以及其他 NLP 应用。

本书的后半部分则将更侧重于介绍如何构建一些 NLP 应用，如对于文本分类，可以用 NLTK 搭配 scikit-learn 库来进行。我们还会讨论一些其他的 Python 库，你应该了解一下这些与文本挖掘或自然语言处理任务相关的库。另外，也会带你看看如何从网页和社交媒体中采集数据，以及如何用 NLTK 进行大规模的文本处理。

本书所涵盖的内容

第 1 章　自然语言处理简介。这一章将会涉及一些 NLP 中的基本概念，并对 NLTK 和 Python 做一些介绍。这一章的重点是让你快速了解 NLTK，并介绍如何安装所需要的库，以便开始构建一个非常基本的单词云实例。

第 2 章　文本的歧义及其清理。这一章将会讨论在任何文本挖掘和 NLP 任务中所需的所有预处理步骤。这一章将会具体讨论断词处理、词干处理、停用词去除等技术。并且，还会为你详细介绍一些别的文本清理技术，以及如何用 NLTK 来简化它们的实现。

第 3 章　词性标注。这一章将重点对词性标注进行概述。在这一章中，我们将会为你介绍如何将 NLTK 运用到一些标注器中，并讨论 NLTK 中有哪些不同的 NLP 标注器可用。

第 4 章　文本结构解析。这一章将会带你继续深入 NLP，讨论不同的语法解析方法，并介绍如何用 NLTK 来实现这些方法。在此过程中，我们会讨论语法解析在 NLP 语境中的，以及一些常见的信息提取技术（如实体提取）中的重要性。

第 5 章　NLP 应用。这一章将会谈及各种不同的 NLP 应用，我们将会带领你利用一些当前已掌握的知识来构建出一个简单的 NLP 应用实例。

第 6 章　文本分类。这一章将会介绍一些机器学习领域中常见的分类方法。讨论重点将主要集中在文本语料库，以及如何用 NLTK 和 scikit 来构建管道，从而实现一个文本分类器。当然，也会讨论与文本聚类和主题模型相关的内容。

第 7 章　Web 爬虫。这一章将讨论 NLP、数据科学和数据收集中其他方面的处理任务，以及如何从最大的文本数据源之一——Web 中获取相关的数据。在这里，我们将学习如何用 Python 库、Scrapy 来建立一只运作良好的 Web 爬虫（crawler）。

第 8 章　NLTK 与其他 Python 库的搭配运用。这一章将会谈及一些骨干的 Python 库，如 NumPy 和 SciPy。另外，我们也会简单地介绍一下用于数据处理的 panda 和用于可视化处理的 matplotlib。

第 9 章　Python 中的社交媒体挖掘。这一章将致力于数据采集相关的内容。在这里，我们将会讨论社交媒体，以及与社交媒体相关的其他问题。当然，我们也会讨论具体应该如何收集、分析并可视化社交媒体中的数据。

第 10 章　大规模文本挖掘。这一章将讨论如何扩展 NLTK，并配合一些别的 Python 库，使其适应大数据时代规模化执行的需要。我们将会给出一个简短的演示，以说明 NLTK 和 scikit 是如何与 Hadoop 搭配使用的。

前期准备

在阅读这本书之前，我们建议你应该准备好下列软件：

章	所需软件（版本）	自由软件/专有软件	软件下载链接	硬件技术指标	所需操作系统
1～5	Python/Anaconda、NLTK	自由软件	https://www.python.org/ http://continuum.io/downloads http://www.nltk.org/	通用UNIX打印系统	不限
6	scikit-learn、gensim	自由软件	http://scikit-learn.org/stable/ https://radimrehurek.com/gensim/	通用UNIX打印系统	不限
7	Scrapy	自由软件	http://scrapy.org/	通用UNIX打印系统	不限
8	NumPy、SciPy、pandas以及matplotlib	自由软件	http://www.numpy.org/ http://www.scipy.org/ http://pandas.pydata.org/ http://matplotlib.org/	通用UNIX打印系统	不限
9	Twitter Python API与Facebook Python API	自由软件	https://dev.twitter.com/overview/api/twitter-libraries https://developers.facebook.com	通用UNIX打印系统	不限

本书的适用读者

只要你是NLP和机器学习领域的爱好者，无论之前有没有文本处理方面的经验，这本书都是为你准备的。当然，这本书也非常适合那些想要快速学习一下NLTK的资深Python程序员。

编写体例

在本书中，我们会用不同的文本样式来突显不同类型信息之间的区别。下面，我们就通过几个例子来介绍一下这些样式，以及它们所代表的含义。

对于正文当中所涉及的代码、数据库表名、文件夹名、文件名、文件扩展名、路径名、

伪 URL、用户输入以及 Twitter 句柄，我们将采取如下形式：“我们需要创建一个名为 NewsSpider.py 文件，并将其路径设置为/tutorial/spiders。”

接下来是 Python 代码块：

```
>>>import nltk
>>>import numpy
```

还有一般性的代码块：

```
add FILE vectorizer.pkl;
add FILE classifier.pkl;
```

另外，在第 7 章中，我们还将会用到 Scrapy shell 中的 IPython 记法，其样式如下：

```
In [1] : sel.xpath('//title/text()')
Out[1]: [<Selector xpath='//title/text()' data=u' Google News'>]
```

最后是所有命令行输入或输出信息的样式：

```
# cp /usr/src/asterisk-addons/configs/cdr_mysql.conf.sample
     /etc/asterisk/cdr_mysql.conf
```

提示：
这种形式表达的是一些需要读者警惕的或需要重点关注的内容。

小技巧：
这种形式所提供的是一些提示或小技巧。

读者反馈

我们始终欢迎任何来自读者的反馈信息。它能让我们了解你对于这本书的看法——无论是喜欢还是不喜欢。这些反馈对于我们的选题开发来说都是至关重要的。

对于一般的反馈，你只需简单地给 feedback@packtpub.com 发一份电子邮件，并在邮件的标题中注明这本书的书名即可。

如果你对某一话题有专长，并且有兴趣写（或奉献）一本这方面的书，请参考我们的作者指南：www.packtpub.com/authors。

客户支持

你一直都是 Packt 图书的主人，我们将会尽一切努力来帮助你获取最好的图书资讯。

实例代码的下载

你可以在 http://www.packtpub.com 自己的账户页面中找到所有已购买的 Packt 图书，并下载相关的实例代码。如果你是在别处购买了我们的图书，也可以通过访问 http://www.packtpub.com/support 注册有关文件，我们会通过电子邮件将其直接发给你。

勘误

尽管我们已经尽了最大的努力来确保书中内容的正确性，但错误始终是存在的。如果你在我们的书中发现了错误——无论是关于文字的还是代码的——只要你能告诉我们，我们都将不胜感激。因为这样可以大大减少其他读者在阅读方面所遇到的困难。因此，当你发现错误时，只需要访问 http://www.packtpub.com/submit-errata，选择相应的书名，然后单击“errata submission form”链接并输入相关错误的详细信息即可。一旦你提供的信息获得了确认，相关的内容就被更新到我们的网站或对应图书勘误章节下面现有的勘误表中。

如果想要查看先前已提交的勘误信息，你只需访问 https://www.packtpub.com/books/content/support，并在其搜索域中输入相关图书的名称，所需信息就会出现在下面的勘误部分中。

版权

在互联网上，版权对于所有媒介而言一直是一个很大的问题。在 Packet，我们向来对于版权许可非常重视。如果你在网络上发现任何形式的我们出版过的作品，都请马上将网址或网站名称告知我们，以便于我们采取补救措施。

请将你怀疑有侵权行为的文档链接发送到：copyright@packetpub.com。

你付出的帮助是对作者权利的保护，我们也由此才能继续为你带来有价值的内容。

如有疑问

如果你对本书有任何疑问，也可以通过 questions@packtpub.com 跟我们联系，我们会竭尽所能地帮你解决问题。

目录

第 1 章
自然语言处理简介

现在，让我们先从介绍自然语言处理（NLP）开始吧。众所周知，语言是人们日常生活的核心部分，任何与语言问题相关的工作都会显得非常有意思。希望这本书能带你领略到 NLP 的风采，并引起学习 NLP 的兴趣。首先，我们需要来了解一下该领域中的一些令人惊叹的概念，并在工作中实际尝试一些具有挑战性的 NLP 应用。

在英语环境中，语言处理研究这一领域通常被简称为 NLP。对语言有深入研究的人通常被叫作语言学家，而“计算机语言学家”这个专用名词则指的是将计算机科学应用于语言处理领域的人。因此从本质上来说，一个计算机语言学家应该既有足够的语言理解能力，同时还可以用其计算机技能来模拟出语言的不同方面。虽然计算机语言学家主要研究的是语言处理理论，但 NLP 无疑是对计算机语言学的具体应用。

NLP 多数情况下指的是计算机上各种大同小异的语言处理应用，以及用 NLP 技术所构建的实际应用程序。在实践中，NLP 与教孩子学语言的过程非常类似。其大多数任务（如对单词、语句的理解，形成语法和结构都正确的语句等）对于人类而言都是非常自然的能力。但对于 NLP 来说，其中有一些任务就必须要转向标识化处理、语块分解、词性标注、语法解析、机器翻译及语音识别等这些领域的一部分，且这些任务有一大部分还仍是当前计算机领域中非常棘手的挑战。在本书中，我们将更侧重于讨论 NLP 的实用方面，因此我们会假设读者在 NLP 上已经有了一些背景知识。所以，读者最好在最低限度上对编程语言有一点了解，并对 NLP 和语言学有一定的兴趣。

在阅读完本章之后，我们希望读者能掌握以下内容。

- 对 NLP 及其相关概念有个基本的了解。
- 完成 Python 和 NLTK 及其他库的安装。
- 编写一些非常基本的 Python 和 NLTK 代码片段。