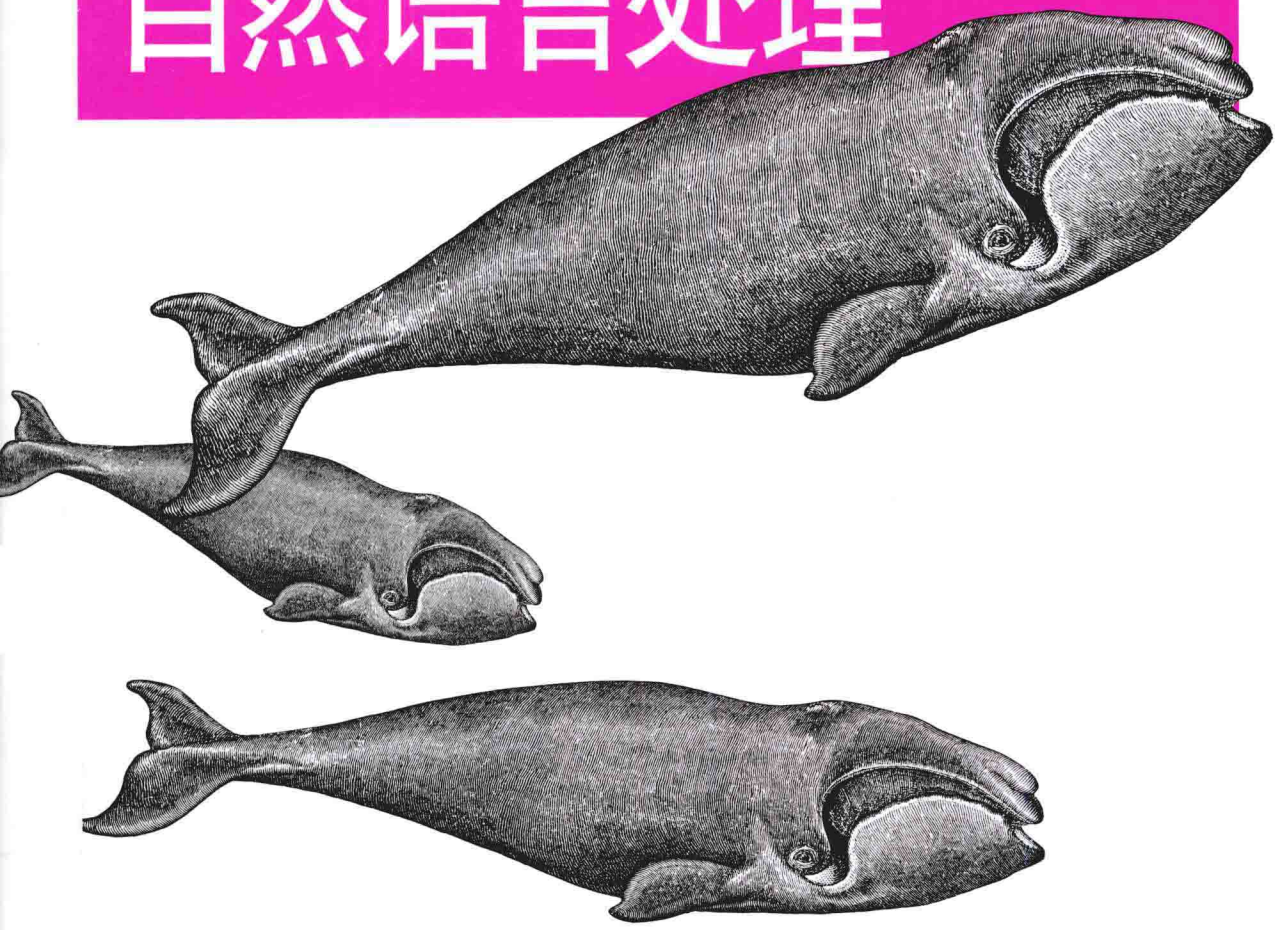
Natural Language Processing with Python

Python
自然语言处理



[美] *Steven Bird, Ewan Klein & Edward Loper* 著
陈涛 张旭 崔杨 刘海平 译

O'REILLY®
人民邮电出版社
POSTS & TELECOM PRESS

O'REILLY®

Python 自然语言处理

[美] Steven Bird Ewan Klein Edward Loper 著

陈 涛 张 旭 崔 杨 刘海平 译

人 民 邮 电 出 版 社

北 京

图书在版编目（CIP）数据

Python自然语言处理 / (美) 伯德 (Bird, S.), (美) 克莱恩 (Klein, E.), (美) 洛佩尔 (Loper, E.) 著 ; 陈涛等译. — 北京 : 人民邮电出版社, 2014.7
ISBN 978-7-115-33368-1

Ⅰ. ①P… Ⅱ. ①伯… ②克… ③洛… ④陈… Ⅲ. ①软件工具－自然语言处理 Ⅳ. ①TP311.56②TP391

中国版本图书馆CIP数据核字(2013)第277137号

◆ 著 [美] Steven Bird Ewan Klein Edward Loper
译 陈 涛 张 旭 崔 杨 刘海平
责任编辑 陈冀康
责任印制 彭志环 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
三河市海波印务有限公司印刷

◆ 开本：787×1000 1/16
印张：31.75
字数：559 千字 2014 年 7 月第 1 版
印数：1-3 000 册 2014 年 7 月河北第 1 次印刷

著作权合同登记号 图字：01-2013-2169 号

定价：89.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315

内容提要

自然语言处理（Natural Language Processing，NLP）是计算机科学领域与人工智能领域中的一个重要方向。它研究实现人与计算机之间用自然语言进行有效通信的各种理论和方法，涉及所有用计算机对自然语言进行的操作。

本书是自然语言处理领域的一本实用入门指南，旨在帮助读者学习如何编写程序来分析书面语言。本书基于 Python 编程语言以及一个名为 NLTK 的自然语言工具包的开源库，但并不要求读者有 Python 编程的经验。全书共 11 章，按照难易程度顺序编排。第 1 章到第 3 章介绍了语言处理的基础，讲述如何使用小的 Python 程序分析感兴趣的文本信息。第 4 章讨论结构化程序设计，以巩固前面几章中介绍的编程要点。第 5 章到第 7 章介绍语言处理的基本原理，包括标注、分类和信息提取等。第 8 章到第 10 章介绍了句子解析、句法结构识别和句意表达方法。第 11 章介绍了如何有效管理语言数据。后记部分简要讨论了 NLP 领域的过去和未来。

本书的实践性很强，包括上百个实际可用的例子和分级练习。本书可供读者用于自学，也可以作为自然语言处理或计算语言学课程的教科书，还可以作为人工智能、文本挖掘、语料库语言学等课程的补充读物。

前言

这是一本关于自然语言处理的书。所谓“自然语言”，是指人们日常交流使用的语言，如英语、印地语、葡萄牙语等。相对于编程语言和数学符号这样的人工语言，自然语言随着一代代的传递而不断演化，因而很难用明确的规则来确定。从广义上讲，“自然语言处理”（Natural Language Processing，NLP）包含所有用计算机对自然语言进行的操作，从最简单的通过计数词汇出现的频率来比较不同的写作风格，到最复杂的完全“理解”人所说的话，或至少达到能对人的话语作出有效反应的程度。

NLP 的技术应用日益广泛。例如：手机和手持电脑对输入法联想提示和手写识别的支持；网络搜索引擎能搜索到非结构化文本中的信息；机器翻译能把中文文本翻译成西班牙文。通过提供更自然的人机界面和获取存储信息的高级手段，语言处理正在这个多语种的信息社会中扮演着更核心的角色。

这本书提供自然语言处理领域的入门指南。它可以用来自学，也可以作为自然语言处理或计算语言学课程的教科书，或是作为人工智能、文本挖掘、语料库语言学课程的补充读物。本书实用性强，包括上百个实例和分级练习。

本书基于 Python 编程语言及名为自然语言工具包（Natural Language Toolkit，NLTK）的开源库。NLTK 包含大量的软件、数据和文档，所有这些都可以从 http://www.nltk.org/上免费下载。NLTK 的发行版本支持 Windows、Macintosh 和 UNIX 平台。强烈建议你下载 Python 和 NLTk，与我们一起尝试书中的例子和练习。

读者

从科学、经济、社会和文化因素来看，NLP 十分重要。NLP 正在迅速成长，其中很多理论和方法在大量新的语言技术中得到广泛应用。所以对很多人来说掌握 NLP 知识十分重要。在应用领域，包括从事人机交互、商业信息分析、Web 软件开发的

人；在学术界，包括从事人文计算学、语料库语言学到计算机科学和人工智能领域的人。（学术界的很多人把 NLP 称为“计算语言学”）

本书旨在帮助所有想要学习编写程序来分析书面语言的人，不管他们以前的编程经验如何。

初学编程？

本书的前几章适合没有编程经验的读者，只要你不怕应对新概念和学习新的计算机技能。书中的例子和数以百计的分级练习，你都可以亲自尝试一下。如果你需要关于 Python 的更一般的介绍，http://docs.python.org/给出了 Python 资源列表。

初学 Python？

有经验的程序员可以很快掌握书中的 Python 代码，而把更多精力专注于自然语言处理。所有涉及的 Python 功能都经过精心解释和举例说明，你很快就会体会到 Python 在这些应用领域的妙用。书中的语言索引会帮你查找书中的相关论述。

已经精通 Python？

你可以浏览一下 Python 的例子并且钻研从第 1 章开始就提到的语言分析材料。很快你就能在这个神奇的领域展现你的技能。

强调

本书是一本介绍 NLP 的实用书籍。你将通过例子学习编写真正的程序，并通过实践验证自己想法的价值。如果你没有学过编程，本书将教你如何**编程**。与其他编程书籍不同的是，我们提供了丰富的 NLP 实例和练习。我们撰写本书时讲究**探究原理**，无论是严谨的语言学还是计算分析学，我们不回避所涉及的任何基础理论。我们曾经试图在理论与实践之间寻求**折中**，确定它们之间的联系与边界。最终我们认识到如果不能寓教于乐，几乎无法实现这个目标，所以我们竭尽所能写入了很多既有益又有趣的应用和例子，有的甚至有些异想天开。

请注意本书并不是一本工具书。本书讲述的 Python 和 NLP 是精心挑选的，并通过

教程的形式展现的。关于参考材料，请查阅 http://python.org/和 http://www.nltk.org/，那里有大量可搜索的资源。

本书也不是高深的计算机科学文章。书中的内容属于入门级和中级，目标读者是那些想要学习如何使用 Python 和自然语言分析包来分析文本的人。若想学习 NLTK 中更高级的算法，你可以查阅 http://www.nltk.org/中的 Python 代码库，或查询在本书中引用的其他文献。

你将学到什么

通过钻研本书，你将学到如下内容：

- 简单的程序如何帮你处理和分析语言数据，以及如何编写这些程序；
- NLP 与语言学的关键概念是如何用来描述和分析语言的；
- 数据结构和算法是怎样在 NLP 中运用的；
- 语言数据如何存储为标准格式，以及如何使用数据来评估 NLP 技术的性能。

根据读者知识背景和学习 NLP 的动机不同，从本书中获得的技能和知识也将不同，详情见表 P-1。

表 P-1　　目标和背景不同的读者，阅读本书可获得的技能和知识

目标	艺术与人文背景	科学与工程背景
语言分析	操控大型语料库，设计语言模型，验证由经验得出的假设	使用数据建模、数据挖掘和知识发掘的技术来分析自然语言
语言技术	应用 NLP 技术构建高效的系统来处理语言学任务	在高效的语言处理软件中使用语言学算法和数据结构

篇章结构

本书前几章按照概念的难易程度编排。先是实用地介绍语言处理，讲述如何使用小的 Python 程序分析感兴趣的文本信息（第 1～3 章）。接着是结构化程序设计章节（第 4 章），用来巩固散布在前面几章中学习的编程要点。之后，加快速度，我们用一系列章节讲述语言处理的基本原理：标注、分类和信息提取（第 5～7 章）。接下来的 3 章探索了句子解析、句法结构识别和句意表达方法构建（第 8～10 章）。

最后一章重点讲述了如何有效管理语言数据（第 11 章）。本书结尾处的后记简要讨论了 NLP 领域的过去和未来。

每一章中我们都在两种不同的叙述风格间切换。一种风格是以自然语言为主线。我们分析语言，探索语言学概念，在讨论中使用编程的例子。我们经常会使用尚未系统介绍的 Python 结构，这样你可以在钻研这些程序如何运作的细节之前了解它们的用处。就像学习一门外语的惯用表达一样，你能够买到好吃的糕点而不必先学会复杂的提问句型。另一种风格是以程序设计语言为主线。我们分析程序、探索算法，而不以语言学例子为主。

每章结尾都有一系列分级练习，用于巩固所学的知识。练习按照如下的标准分级：○初级练习，对范例代码稍加修改等简单的练习；◑中级练习，深入探索所学知识的某个方面，需要仔细地分析和设计；●高级练习，开放式的任务，挑战你对所学知识的理解并要求你独立思考解决的方案（新学编程的读者可以跳过这些）。

每一章都有深入阅读环节和放置在 http://www.nltk.org/网站上的“额外内容”部分，用来介绍更深入的相关材料及一些网络资源。所有实例代码都可从网上下载。

为什么使用 Python？

Python 是一种简单但功能强大的编程语言，其自带的函数非常适合处理语言数据。Python 可以从 http://www.python.org/免费下载，并能够在各种平台上安装运行。

下面的 4 行 Python 程序就可以操作 file.txt 文件，输出所有后缀是“ing”的词。

```
>>> for line in open("file.txt"):
... for word in line.split():
... if word.endswith('ing'):
... print word
```

这段程序演示了 Python 的一些主要特征。第一，使用空白符号缩进代码，从而使 if 后面的代码都在前面一行 for 语句的范围之内；这能保证对每个单词都能进行“ing”结尾检测。第二，Python 是面向对象语言。每一个变量都是包含特定属性和方法的实例。例如：变量“line”的值不仅仅是一行字符串，它是一个 string 对象，包含用来把字符串分割成词的 split()方法（或叫操作、函数）。我们在对象名称后添加句号（点）和方法名称就可以调用对象的一个方法，即 line.split()。第三，

方法的参数写在括号内。例如：上面例子中的 word.endswith('ing')，参数“ing”表示我们需要找的是以“ing”结尾的词而不是别的结尾的词。最后也是最重要的，Python 的可读性非常强可以很容易地猜出程序的功能，即使你以前从未写过一行代码。

选择 Python 是因为它的学习曲线比较平缓，文法和语义都很易懂，具有强大的字符串处理功能。作为解释性语言，Python 便于交互式编程。作为面向对象语言，Python 允许数据和方法被方便地封装和重用。作为动态语言，Python 允许属性在等到程序运行时添加到对象，允许变量自动类型转换，提高开发效率。Python 自带强大的标准库，包括图形编程、数值处理和网络连接等组件。

Python 在世界各地的工业、科研、教育领域应用广泛，因其提高了软件的生产效率、质量和可维护性而备受欢迎。http://www.python.org/about/success/列举了许多成功使用 Python 的例子。

NLTK 定义了使用 Python 进行 NLP 编程的基础工具。它提供了与自然语言处理相关的数据表示基本类，词性标注、文法分析、文本分类等任务的标准接口及这些任务的标准实现，可以组合起来解决复杂的问题。

NLTK 自带大量文档。作为本书的补充，http://www.nltk.org/网站提供的 API 文档涵盖了工具包中每一个模块、类和函数，详细说明了各种参数，以及用法示例。该网站还为广大用户、开发人员和导师提供了很多包含大量的例子和测试用例的 HOWTO。

软件安装需求

为了充分利用好本书，你应该安装一些免费的软件包。http://www.nltk.org/上有这些软件包当前的下载链接和安装说明。

Python

本书的例子都假定你正在使用 Python 2.4 或 2.5 版本。一旦 NLTK 依赖的库支持 Python 3.0，我们将把 NLTK 移植到 Python 3.0。

NLTK

本书的代码示例使用 NLTK 2.0 版本。NLTK 的后续版本是兼容的。

NLTK-Data

包含本书中所分析和处理的语言语料库。

NumPy（推荐）

这是一个科学计算库，支持多维数组和线性代数，在某些概率计算、标记、聚类和分类任务中会用到。

Matplotlib（推荐）

这是一个用于数据可视化的 2D 绘图库，在产生线图和条形图的程序例子中会用到。

NetworkX（可选）

这是一个用于存储和操作由节点和边组成的网络结构的函数库。实现可视化语义网络还需要安装 Graphviz 库。

Prover9（可选）

这是一个使用一阶等式逻辑的定理自动证明器，用于支持语言处理中的推理。

自然语言工具包（NLTK）

NLTK 创建于 2001 年，最初是宾州大学计算机与信息科学系计算语言学课程的一部分。从那以后，在数十名贡献者的帮助下不断发展壮大。如今，它已被几十所大学的课程所采纳，并作为许多研究项目的基础。表 P-2 列出了 NLTK 的一些最重要的模块。

表 P-2　　语言处理任务与相应 NLTK 模块及功能描述

语言处理任务	NLTK 模块	功能描述
获取语料库	nltk.corpus	语料库和词典的标准化接口
字符串处理	nltk.tokenize, nltk.stem	分词、句子分解、提取主干
搭配探究	nltk.collocations	t-检验、卡方、点互信息
词性标识符	nltk.tag	n-gram、backoff、Brill、HMM、TnT
分类	nltk.classify, nltk.cluster	决策树、最大熵、朴素贝叶斯、EM、k-means
分块	nltk.chunk	正则表达式、n-gram、命名实体
解析	nltk.parse	图表、基于特征、一致性、概率性、依赖项
语义解释	nltk.sem, nltk.inference	λ 演算、一阶逻辑、模型检验
指标评测	nltk.metrics	精度、召回率、协议系数
概率与估计	nltk.probability	频率分布、平滑概率分布
应用	nltk.app, nltk.chat	图形化的关键词排序、分析器、WordNet 查看器、聊天机器人
语言学领域的工作	nltk.toolbox	处理 SIL 工具箱格式的数据

NLTK 设计中的 4 个主要目标如下。

简易性

提供直观的框架和大量模块，使用户获取 NLP 知识而不必陷入像标注语言数据那样繁琐的事务中。

一致性

提供具有一致的接口和数据结构并且方法名称容易被猜到的统一框架。

可扩展性

提供一种结构使得新的软件模块可以方便添加进来，模块包括同一任务中不同的或相互冲突的实现方式。

模块化

提供可以独立使用而与工具包的其他部分无关的组件。

对比上述目标，我们回避了工具包的潜在实用性。首先，虽然工具包提供了广泛的工具，但它不是面面俱全的。第一，它是一个工具包而不是一个系统，它将会随着 NLP 领域一起发展。第二，虽然这个工具包的效率足以支持实际的任务，但它运行时的性能还没有高度优化。这种优化往往涉及更复杂的算法或使用 C 或 C++等较低一级的编程语言来实现。这将使得工具包的可读性变差且更难以安装。第三，我们试图避开巧妙的编程技巧，因为我们相信清楚直白的实现比巧妙却可读性差的方法好。

对老师的话

自然语言处理一般是在高年级本科生或研究生阶段开设的为期一个学期的课程。很多教师都发现，在如此短的时间里涵盖理论和实践两个方面是十分困难的。有些课程注重理论而排除掉实践练习，这剥夺了学生编写程序自动处理语言带来的挑战和兴奋感。另一些课程仅仅教授语言学编程而不包含任何重要的 NLP 内容。最初开发 NLTK 就是为了解决这个问题，无论学生之前是否具有编程经验，都能使教师在一个学期里同时教授大量理论和实践成为可能。

在所有 NLP 教学大纲中算法和数据结构部分都十分重要。它们本身可能非常枯燥，而 NLTK 提供的交互式图形用户界面能让读者一步一步看到算法过程，使它们变得鲜活。大多数 NLTK 组件都有一个无需用户输入任何数据就能执行有趣任务的示范性例子。学习本书的一种有效方法就是通过交互式重现书中的范例，把它们输入到 Python 会话控制台，观察它们的功能，尝试修改它们去探索经验性问题或者理论性问题。

本书包含了数百个练习，可作为学生作业。最简单的练习包括用指定的方式修改已有的程序片段来回答具体的问题。另一方面，NLTK 为研究生水平的研究项目提供了一个灵活的框架，包括所有的基本数据结构和算法的标准实现，几十个广泛使用的数据集（语料库）的接口，以及一个灵活可扩展的体系结构。NLTK 网站上还有其他资源可以支持 NLTK 教学。

我们相信本书是唯一能为学生提供在学习编程的环境中学习 NLP 的综合性教程。各个章节和练习通过与 NLTK 紧密结合，并将各章材料有序分割开，为学生（即使是那些以前没有编程经验的学生）提供一个实用的 NLP 的入门指南。学完这些材料后，学生能准备好尝试一本更加深层次的教科书，例如：Speech and Language Processing（《语音和语言处理》），作者是 Jurafsky 和 Martin（Prentice Hall 出版社，2008 年）。

本书介绍编程概念的顺序与众不同。以一个重要的数据类型：字符串列表（链表）

开始，然后介绍重要的控制结构，如推导和条件式等。这些常用知识允许我们在一开始就做一些有用的语言处理。当有了这样动机，我们再回过头来系统地介绍一些基础概念，如字符串、循环、文件等。这种方法同更传统的方法相比，达到了同样的效果而不必要求读者对编程感兴趣。

表 P-3 列出了两个课程计划表。第一个适用于艺术人文专业背景的读者，第二个适用于科学与工程背景的读者。其他的课程计划应该涵盖前 5 章，然后把剩余的时间投入单独的领域，例如：文本分类（第 6、7 章）、文法（第 8、9 章）、语义（第 10 章）或者语言数据管理（第 11 章）。

表 P-3　　课程计划建议（每一章近似的课时数）

章	艺术人文专业	理工科
第 1 章　语言处理与 Python	2～4	2
第 2 章　获得文本语料和词汇资源	2～4	2
第 3 章　处理原始文本	2～4	2
第 4 章　编写结构化程序	2～4	1～2
第 5 章　分类和标注词汇	2～4	2～4
第 6 章　学习分类文本	0～2	2～4
第 7 章　从文本提取信息	2	2～4
第 8 章　分析句子结构	2～4	2～4
第 9 章　建立基于特征的文法	2～4	1～4
第 10 章　分析语句的含义	1～2	1～4
第 11 章　语言数据管理	1～2	1～4
总计	18～36	18～36

本书使用的约定

本书使用以下印刷约定。

黑体

表示新的术语。

斜体

用在段落中表示语言学例子、文本的名称和 URL，文件名和后缀名也用斜体。

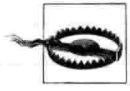
等宽字体

用来表示程序清单，用在段落中表示变量、函数名、声明或关键字等程序元素。也用来表示程序名。

等宽斜体

表示应由用户提供的值或上下文决定的值来代替文本中的值，也在程序代码例子中用来表示元变量。

这个图标表示提示、建议或一般性的提醒。



这个图标表示警告。

使用代码范例

本书是为了帮你完成工作的。一般情况下，你可以在你的程序或文档中使用本书中的代码，而不需要得到我们的允许，当你需要大量地复制代码时除外。例如，编写程序时用到书中的几段代码不需要许可。销售和分发包含 O'Reilly 书籍中例子的 CD-ROM 需要获得许可。援引本书和书中例子来回答问题不需要许可。大量地将本书中的例子纳入你的产品文档将需要获得许可。

我们希望但不强求被参考文献引用。引用通常包括标题、作者、出版者和 ISBN。例如：“Natural Language Bocessing with Rthow，Steven Bird，Ewan Klein 和 Edward Loper。版权所有 2009 Steven Bird, Ewan Klein 和 Edward Loper, 978-0-596-51649-9。”如果你觉得你使用本书的例子代码超出了上面列举的一般用途或许可，请通过 permissions@oreilly.
com 随时联系我们。

Safari®在线丛书

当你看到任何你喜爱的技术书的封面上印有 Safari®在线丛书的图标时，这意味着这本书可以在 O'Reilly 网络 Safari 书架上找到。

Safari 提供比电子书更好的解决方案。它是一个虚拟图书馆，你可以轻松搜索数以千计的顶尖技术书籍，可剪切和粘贴例子代码，并下载一些章节，在你需要最准确最新的信息时快速找到答案。欢迎免费试用 http://my.safaribooksonline.com。

如何联系我们

关于本书的意见和咨询请写信给出版商。

O'Reilly Media 公司
1005 Gravenstein Highway North
Sebastopol, CA 95472
中国：
北京市西城区西直门南大街 2 号成铭大厦 C 座 807（100035）
奥莱利技术咨询（北京）有限公司

我们为本书的勘误表、例子等信息制作了一个网页。你可以访问这个页面：

http://www.oreilly.com/catalog/9780596516499

作者通过 NLTK 网站提供了各章的其他材料：

http://www.nltk.org/

要发表评论或询问有关这本书的技术问题，发送电子邮件至：

bookquestions@oreilly.com

欲了解更多有关我们的书籍、会议、资源中心和 O'Reilly 网络的信息，请参阅我们的网站：

http://www.oreilly.com

致谢

作者感激为本书早期手稿提供反馈意见的专家，他们是：Doug Arnold、Michaela Atterer、Greg Aumann、Kenneth Beesley、Steven Bethard、Ondrej Bojar、Chris Cieri、Robin Cooper、Grev Corbett、James Curran、Dan Garrette、Jean Mark Gawron、Doug Hellmann、Nitin Indurkhya、Mark Liberman、Peter Ljunglöf、Stefan Müller、Robin Munn、Joel Nothman、Adam Przepiorkowski、Brandon Rhodes、Stuart Robinson、Jussi Salmela、Kyle Schlansker、Rob Speer 和 Richard Sproat。感谢学生和同事们，他们对课堂材料的宝贵意见演化成本

书的相关章节，其中包括巴西、印度和美国的 NLP 与语言学暑期学校的参加者。没有 NLTK 开发社区的成员的努力就不会产生这本书，他们为建设和壮大 NLTK 无私奉献了他们的时间和专业知识，他们的名字都记录在 NLTK 网站上。

非常感谢美国国家科学基金会、语言数据联盟、Edward Clarence Dyason 奖学金、宾州大学、爱丁堡大学和墨尔本大学对本书相关工作的支持。

感谢 Julie Steele、Abby Fox、Loranah Dimant 及其他 O'Reilly 团队成员。他们组织大量 NLP 和 Python 社区成员全面审阅我们的手稿，还主动为满足我们的需要而定制 O'Reilly 的生成工具。感谢他们一丝不苟的审稿工作。

最后，深深地感谢我们的合伙人，他们是 Kay、Mimo 和 Jee。感谢在我们写作本书的几年里他们付出的关心、耐心和支持。我们希望我们的孩子——Andrew、Alison、Kirsten、Leonie 和 Maaike——能从这些页面中感觉到我们对语言和计算的热情。

版税

这本书的版税将用来支持自然语言工具包的发展。

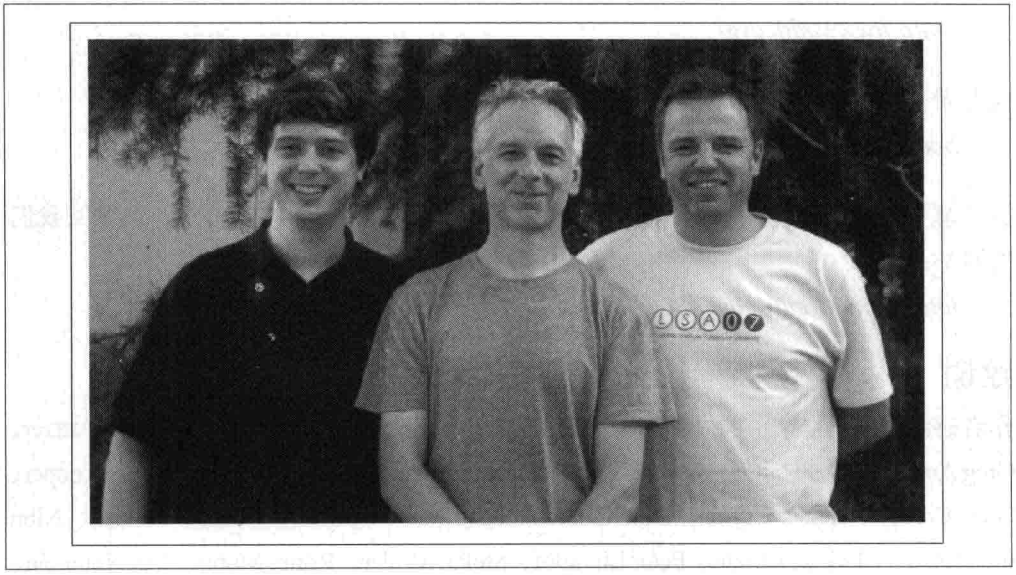


图 P-1. Edward Loper、Ewan Klein 和 Steven Bird，斯坦福大学，2007 年 7 月

作者简介

Steven Bird 是墨尔本大学计算机科学和软件工程系副教授，宾夕法尼亚大学的语言数据联盟高级副研究员。他于 1990 年在英国爱丁堡大学完成计算音韵学博士，导师是 Ewan Klein。后来到喀麦隆开展夏季语言学研究所主持的 Grassfields 班图语语言实地调查。最近，他作为语言数据联盟副主任带领研发队伍花了几年时间，创建已标注文本的大型数据库的模型和工具。在墨尔本大学，他建立了一个语言技术研究组，并在各级本科计算机科学课程任教。2009 年，史蒂芬成为计算语言学学会主席。

Ewan Klein 是英国爱丁堡大学信息学院语言技术教授。于 1978 年在剑桥大学完成形式语义学博士学位。在苏塞克斯和纽卡斯尔大学工作多年后，开始在爱丁堡从事教学工作。于 1993 年他参与了爱丁堡语言科技集团的建立，并一直与之密切联系。从 2000 年到 2002 年，他离开大学，在圣克拉拉的埃迪法公司的总部——爱丁堡的自然语言的研究小组担任研发经理，负责处理口语对话。Ewan 是欧洲章计算语言学协会（European Chapter of the Association for Computational Linguistics）前任主席，并且是人类语言技术（ELSNET）欧洲卓越网络的创始成员和协调员。

Edward Loper 最近完成了宾夕法尼亚大学自然语言处理的机器学习博士学位。爱德华是史蒂芬在 2000 年秋季计算语言学研究生课程的学生，也是教师助手和 NLTK 开发的成员。除了 NLTK，他帮助开发了用于记录和测试 Python 软件的两个包：epydoc 和 doctest。