Packt>

异步图书
www.epubit.com.cn

用Python开发令人惊讶的NLP项目

精通 Python 自然语言处理

Mastering Natural Language Processing with Python

[印度] Deepti Chopra　Nisheeth Joshi　Iti Mathur 著
王威 译

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

精通 Python 自然语言处理

[印度] Deepti Chopra　Nisheeth Joshi　Iti Mathur 著
王威 译

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

精通Python自然语言处理 / (印) 乔普拉 (Deepti Chopra) , (印) 乔希 (Nisheeth Joshi) , (印) 摩突罗 (Iti Mathur) 著 ; 王威译. -- 北京 : 人民邮电出版社, 2017.8
ISBN 978-7-115-45968-8

Ⅰ. ①精… Ⅱ. ①乔… ②乔… ③摩… ④王… 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(2017)第153438号

版权声明

◆ 著　[印度] Deepti Chopra　Nisheeth Joshi　Iti Mathur
译　王　威
责任编辑　陈冀康
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：14
字数：274 千字　2017 年 8 月第 1 版
印数：1－2 400 册　2017 年 8 月北京第 1 次印刷

著作权合同登记号　图字：01-2017-4814 号

定价：59.00 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：京东工商广登字 20170147 号

内容提要

自然语言处理是计算语言学和人工智能之中与人机交互相关的领域之一。

本书是学习自然语言处理的一本综合学习指南，介绍了如何用 Python 实现各种 NLP 任务，以帮助读者创建基于真实生活应用的项目。全书共 10 章，分别涉及字符串操作、统计语言建模、形态学、词性标注、语法解析、语义分析、情感分析、信息检索、语篇分析和 NLP 系统评估等主题。

本书适合熟悉 Python 语言并对自然语言处理开发有一定了解和兴趣的读者阅读参考。

作者简介

Deepti Chopra 是 Banasthali 大学的助理教授。她的主要研究领域是计算语言学、自然语言处理以及人工智能，她也参与了将英语转换为印度诸语言的机器翻译引擎的研发。她在各种期刊和会议上发表过一些文章，此外她还担任一些期刊及会议的程序委员会委员。

Nisheeth Joshi 是 Banasthali 大学的副教授。他感兴趣的领域包括计算语言学、自然语言处理以及人工智能。除此之外，他也非常积极地参与了将英语转换为印度诸语言的机器翻译引擎的研发。他是印度政府电子和信息技术部 TDIL 计划选任的专家之一，TDIL 是负责印度语言技术资金和研究的主要组织。他在各种期刊和会议上发表过一些文章，并同时担任一些期刊及会议的程序委员会及编审委员会委员。

Iti Mathur 是 Banasthali 大学的助理教授。她感兴趣的领域是计算语义和本体工程。除此之外，她也非常积极地参与了将英语转换为印度诸语言的机器翻译引擎的研发。她是印度政府电子和信息技术部 TDIL 计划选任的专家之一，TDIL 是负责印度语言技术资金和研究的主要组织。她在期刊和会议上发表过一些文章，并同时担任一些期刊及会议的程序委员会及编审委员会委员。

我们要诚挚地感谢所有的亲朋好友，因为你们的祝福促使我们完成了出版这本基于自然语言处理的图书的目标。

审阅者简介

Arturo Argueta 目前是一名在读博士研究生，他专注于高性能计算和自然语言处理领域的研究。他在聚类算法、有关自然语言处理的机器学习算法以及机器翻译等方面有一定的研究。他还精通英语、德语和西班牙语。

译者简介

王威　资深研发工程师，曾就职于携程、东方财富等互联网公司。目前专注于互联网分布式架构设计、大数据与机器学习、算法设计等领域的研究，擅长 C#、Python、Java、C++等技术。内涵段子手、空想创业家、业余吉他手、重度读书人。

前言

在本书中，我们将学习如何使用 Python 实现各种有关自然语言处理的任务，并了解一些有关自然语言处理的当下和新进的研究主题。本书是一本综合的进阶指南，以期帮助学生和研究人员创建属于他们自己的基于真实生活应用的项目。

本书涵盖内容

第 1 章，字符串操作，介绍如何执行文本上的预处理任务，例如切分和标准化，此外还介绍了各种字符串匹配方法。

第 2 章，统计语言建模，包含如何计算单词的频率以及如何执行各种语言建模的技术。

第 3 章，形态学：在实践中学习，讨论如何开发词干提取器、形态分析器以及形态生成器。

第 4 章，词性标注：单词识别，解释词性标注以及有关 n-gram 方法的统计建模。

第 5 章，语法解析：分析训练资料，提供关于 Tree bank 建设、CFG 建设、CYK 算法、线图分析算法以及音译等概念的相关信息。

第 6 章，语义分析：意义很重要，介绍浅层语义分析（即 NER）的概念和应用以及使用 Wordnet 执行 WSD。

第 7 章，情感分析：我很快乐，提供可以帮助你理解和应用情感分析相关概念的信息。

第 8 章，信息检索：访问信息，将帮助你理解和应用信息检索及文本摘要的概念。

第 9 章，语篇分析：理解才是可信的，探讨语篇分析系统和基于指代消解的系统。

第 10 章，NLP 系统评估：性能分析，谈论 NLP 系统评估相关概念的理解与应用。

本书的阅读前提

本书中所有的代码示例均使用 Python 2.7 或 Python 3.2 以上的版本编写。不管是 32 位机还是 64 位机，都必须安装 NLTK（Natural Language Toolkit，NLTK）3.0 包。操作系统要求为 Windows、Mac 或 UNIX。

本书的目标读者

本书主要面向对 Python 语言有一定认知水平的自然语言处理的中级开发人员。

排版约定

本书中用不同的文本样式来区分不同种类的信息。下面给出了这些文本样式的示例及其含义。

文本中的代码单词、数据库表名、文件夹名称、文件名、文件扩展名、路径名、虚拟 URL、用户输入以及推特用户定位表示如下：

“对于法语文本的切分，我们将使用 `french.pickle` 文件。”

代码块的样式如下所示：

```
>>> import nltk
>>> text=" Welcome readers. I hope you find it interesting. Please do
reply."
>>> from nltk.tokenize import sent_tokenize
```

此图标表示警告或需要特别注意的内容。

此图标表示提示或者技巧。

读者反馈

我们始终欢迎来自读者的反馈。请告诉我们你对本书的看法——喜欢或者不喜欢的部分。你的意见对我们来说非常重要，这将有助于我们开发出读者真正感兴趣的东西。

一般的反馈，你只需发送邮件至 feedback@packtpub.com，并在邮件主题中写清楚书名。

如果你擅长某个主题，并有兴趣编写一本书或者想为一本书做贡献，请参考我们的作者指南，网址 www.packtpub.com/authors。

客户支持

既然你已经是 Packt 引以为傲的读者了，为了能让你的购买物超所值，我们还为你准备了以下内容。

下载示例代码

你可以用你的 http://www.packtpub.com 账户在上面下载本书配套的示例代码。如果你是在别的地方购买的本书，你可以访问 http://www.packtpub.com/support 并注册，我们会用邮件把代码文件直接发给你。

你可以按照以下步骤下载代码文件。

1. 使用你的邮箱地址和密码登录或注册我们的网站。
2. 将鼠标指针移至顶端的 SUPPORT 选项卡上。
3. 单击 Code Downloads & Errata。
4. 在搜索框中输入书名。
5. 选择你需要下载代码文件的图书。
6. 在下拉菜单里选择你从哪里购买的这本书。
7. 单击 Code Download。

你也可以通过单击 Packt 出版社官网上关于本书的网页中的“Code Files”按钮来下载代码文件。你可以通过在搜索框中输入书名进入到这个页面。请注意你需要登录你的 Packt 账户。

一旦下载示例代码文件后，请确保使用以下最新版本的工具解压文件夹：

- WinRAR / 7-Zip for Windows。
- Zipeg / iZip / UnRarX for Mac。
- 7-Zip / PeaZip for Linux。

本书的代码包也托管在 Github 上，网址是 `https://github.com/PacktPublishing/Mastering-Natural-Language-Processing-with-Python`。我们也有来自于我们丰富的图书和视频目录的其他代码包，地址是 `https://github.com/PacktPublishing/`。欢迎访问！

勘误

虽然我们竭尽全力保证图书内容的准确性，但错误仍在所难免。如果你在我们的任何一本书里发现错误，可能是文字的或者代码中的错误，都烦请报告给我们，我们将不胜感激。这样不仅使其他读者免于困惑，也能帮助我们不断改进后续版本。如果你发现任何错误，请访问 `http://www.packtpub.com/submit-errata` 报告给我们，选择相应图书，单击“Errata Submission Form”链接，并输入勘误详情。一旦你提出的错误被证实，你的勘误将被接收并上传至我们的网站，或加入到已有的勘误列表中。

若要查看之前提交的勘误，请访问 `https://www.packtpub.com/books/ content/support` 并在搜索框中输入书名，所需的信息将会展现在“Errata”部分的下面。

反盗版

在互联网上，所有媒体都会遭遇盗版问题。对 Packt 来说，我们严格保护版权和许可证。如果你在互联网上发现我们出版物的任何非法副本，请立即向我们提供侵权网站的地址和名称，以便我们采取补救措施。

请通过 copyright@packtpub.com 联系我们，同时请提供涉嫌侵权内容的链接。

非常感激你帮助保护我们的作者，让我们尽力提供更有价值的内容。

问题

如果你对本书有任何疑问，都可以通过 questions@packtpub.com 邮箱联系我们，我们将尽最大努力为你答疑解惑。

目录

第 1 章
字符串操作

自然语言处理（Natural Language Processing，NLP）关注的是自然语言与计算机之间的交互。它是人工智能（Artificial Intelligence，AI）和计算语言学的主要分支之一。它提供了计算机和人类之间的无缝交互并使得计算机能够在机器学习的帮助下理解人类语言。在编程语言（例如 C、C++、Java、Python 等）里用于表示一个文件或文档内容的基础数据类型被称为字符串。在本章中，我们将探索各种可以在字符串上执行的操作，这些操作将有助于完成各种 NLP 任务。

本章将包含以下主题：

- 文本切分。
- 文本标准化。
- 替换和校正标识符。
- 在文本上应用 Zipf 定律。
- 使用编辑距离算法执行相似性度量。
- 使用 Jaccard 系数执行相似性度量。
- 使用 Smith Waterman 算法执行相似性度量。

1.1 切分

切分可以认为是将文本分割成更小的并被称作标识符的模块的过程，它被认为是 NLP 的一个重要步骤。

当安装好 NLTK 包并且 Python 的交互式开发环境（IDLE）也运行起来时，我们就可以将文本或者段落切分成独立的语句。为了实现切分，我们可以导入语句切分函数，该函数的参数即为需要被切分的文本。sent_tokenize 函数使用了 NLTK 包的一个叫作 PunktSentenceTokenizer 类的实例。基于那些可以标记句子开始和结束的字母和标点符号，NLTK 中的这个实例已经被训练用于对不同的欧洲语言执行切分。

1.1.1 将文本切分为语句

现在，让我们来看看一段给定的文本是如何被切分为独立的句子的：

```
>>> import nltk
>>> text=" Welcome readers. I hope you find it interesting. Please do
reply."
>>> from nltk.tokenize import sent_tokenize
>>> sent_tokenize(text)
[' Welcome readers.', 'I hope you find it interesting.', 'Please do
reply.']
```

这样，一段给定的文本就被分割成了独立的句子。我们还可以进一步对这些独立的句子进行处理。

要切分大批量的句子，我们可以加载 PunktSentenceTokenizer 并使用其 tokenize()函数来进行切分。下面的代码展示了该过程：

```
>>> import nltk
>>> tokenizer=nltk.data.load('tokenizers/punkt/english.pickle')
>>> text=" Hello everyone. Hope all are fine and doing well. Hope you
find the book interesting"
>>> tokenizer.tokenize(text)
[' Hello everyone.', 'Hope all are fine and doing well.', 'Hope you
find the book interesting']
```

1.1.2 其他语言文本的切分

为了对除英文之外的其他语言执行切分，我们可以加载它们各自的 pickle 文件（可以在 tokenizers/punkt 里边找到），然后用该语言对文本进行切分，这些文本是 tokenize()函数的参数。对于法语文本的切分，我们将使用如下的 french.pickle 文件：

```
>>> import nltk
>>> french_tokenizer=nltk.data.load('tokenizers/punkt/french.pickle')
>>> french_tokenizer.tokenize('Deux agressions en quelques jours,
```

```
voilà ce qui a motivé hier matin le débrayage collège franco-
britanniquede Levallois-Perret. Deux agressions en quelques jours,
voilà ce qui a motivé hier matin le débrayage Levallois. L'équipe
pédagogique de ce collège de 750 élèves avait déjà été choquée
par l'agression, janvier , d'un professeur d'histoire. L'équipe
pédagogique de ce collège de 750 élèves avait déjà été choquée par
l'agression, mercredi , d'un professeur d'histoire')
['Deux agressions en quelques jours, voilà ce qui a motivé hier
matin le débrayage collège franco-britanniquedeLevallois-Perret.',
'Deux agressions en quelques jours, voilà ce qui a motivé hier matin
le débrayage Levallois.', 'L'équipe pédagogique de ce collège de
750 élèves avait déjà été choquée par l'agression, janvier , d'un
professeur d'histoire.', 'L'équipe pédagogique de ce collège de
750 élèves avait déjà été choquée par l'agression, mercredi , d'un
professeur d'histoire']
```

1.1.3 将句子切分为单词

现在，我们将对独立的句子执行处理，独立的句子会被切分为单词。通过使用word_tokenize()函数可以执行单词的切分。word_tokenize 函数使用 NLTK 包的一个叫作 TreebankWordTokenizer 类的实例用于执行单词的切分。

使用 word_tokenize 函数切分英文文本的代码如下所示：

```
>>> import nltk
>>> text=nltk.word_tokenize("PierreVinken , 59 years old , will join
as a nonexecutive director on Nov. 29 .»)
>>> print(text)
['PierreVinken', ',', '59', ' years', ' old', ',', 'will', 'join',
'as', 'a', 'nonexecutive', 'director' , 'on', 'Nov.', '29', '.']
```

实现单词的切分还可以通过加载 TreebankWordTokenizer，然后调用 tokenize()函数来完成，其中 tokenize()函数的参数是需要被切分为单词的句子。基于空格和标点符号，NLTK 包的这个实例已经被训练用于将句子切分为单词。

如下代码将帮助我们获取用户的输入，然后再将其切分并计算切分后的列表长度：

```
>>> import nltk
>>> from nltk import word_tokenize
>>> r=input("Please write a text")
Please write a textToday is a pleasant day
>>> print("The length of text is",len(word_tokenize(r)),"words")
The length of text is 5 words
```