



# Java自然语言处理 (影印版)

Natural Language Processing with Java

Richard M Reese 著

[PACKT]  
PUBLISHING



东南大学出版社  
SOUTHEAST UNIVERSITY PRESS

# Java 自然语言处理(影印版)

*Richard M Reese* 著

南京 东南大学出版社

## 图书在版编目(CIP)数据

Java 自然语言处理:英文/(英)里斯(Reese, R.M.)  
著. —影印本. —南京:东南大学出版社, 2016.1

书名原文: Natural Language Processing with Java

ISBN 978-7-5641-6088-3

I. ①J… II. ①里… III. ①JAVA 语言—程序设计—英文②自然语言处理—英文 IV. ①TP312②TP391

中国版本图书馆 CIP 数据核字(2015)第 256587 号

© 2015 by PACKT Publishing Ltd

Reprint of the English Edition, jointly published by PACKT Publishing Ltd and Southeast University Press, 2016. Authorized reprint of the original English edition, 2015 PACKT Publishing Ltd, the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 PACKT Publishing Ltd 出版 2015。

英文影印版由东南大学出版社出版 2016。此影印版的出版和销售得到出版权和销售权的所有者——PACKT Publishing Ltd 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

## Java 自然语言处理(影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出 版 人: 江建中

网 址: <http://www.seupress.com>

电子邮件: [press@seupress.com](mailto:press@seupress.com)

印 刷: 常州市武进第三印刷有限公司

开 本: 787 毫米×980 毫米 16 开本

印 张: 16.25

字 数: 318 千字

版 次: 2016 年 1 月第 1 版

印 次: 2016 年 1 月第 1 次印刷

书 号: ISBN 978-7-5641-6088-3

定 价: 56.00 元

本社图书若有印装质量问题, 请直接与营销部联系。电话(传真): 025-83791830

# Credits

**Author**

Richard M Reese

**Reviewers**

Suryaprakash CV

Evan Dempsey

Anil Omanwar

Amitabh Sharma

**Commissioning Editor**

Nadeem N. Bagban

**Acquisition Editor**

Sonali Vernekar

**Content Development Editor**

Ritika Singh

**Technical Editor**

Manali Gonsalves

**Copy Editors**

Pranjali Chury

Vikrant Phadke

**Project Coordinators**

Aboli Ambardekar

Judie Jose

**Proofreaders**

Simran Bhogal

Jonathan Todd

**Indexer**

Priya Sane

**Production Coordinator**

Nitesh Thakur

**Cover Work**

Nitesh Thakur

# About the Author

**Richard M Reese** has worked in both industry and academics. For 17 years, he worked in the telephone and aerospace industries, serving in several capacities, including research and development, software development, supervision, and training. He currently teaches at Tarleton State University, where he is able to apply his years of industry experience to enhance his classes.

Richard has written several Java and C books. He uses a concise and easy-to-follow approach to topics at hand. His books include *EJB 3.1 Cookbook*; books about new features of Java 7 and 8, Java Certification, and jMonkey Engine; and a book on C pointers.

---

I would like to thank my daughter, Jennifer, for the numerous reviews and contributions she has made. Her input has been invaluable.

---

# About the Reviewers

**Suryaprakash C.V.** has been working in the field of NLP since 2009. He has done his graduation in physics and postgraduation in computer applications. Later, he got an opportunity to pursue a career in his area of interest, which is natural language processing.

Currently, Suryaprakash is a research lead at Senseforth Technologies.

---

I would like to thank my colleagues for supporting me in my career and job. It helped me a lot in this review process.

---

**Evan Dempsey** is a software developer from Waterford, Ireland. When he isn't hacking using Python for fun and profit, he enjoys craft beers, Common Lisp, and keeping up with modern research in machine learning. He is a contributor to several open source projects.

**Anil Omanwar** is a dynamic personality with a great passion for the hottest technology trends and research. He has more than 8 years of experience in researching cognitive computing. Natural language processing, machine learning, information visualization, and text analytics are a few key areas of his research interests.

He is proficient in sentiment analysis, questionnaire-based feedback, text clustering, and phrase extraction in diverse domains, such as life sciences, manufacturing, retail, e-commerce, hospitality, customer relations, banking, and social media.

Anil is currently associated with IBM labs for NLP and IBM Watson in the life sciences domain. The objective of his research is to automate critical manual steps and assist domain experts in optimizing human-machine capabilities.

In his spare time, he enjoys working for social causes, trekking, photography, and traveling. He is always ready to take up technical challenges.

**Amitabh Sharma** is a professional software engineer. He has worked extensively on enterprise applications in telecommunications and business analytics. His work has focused on service-oriented architecture, data warehouses, and languages such as Java, Python, and so on.

# www.PacktPub.com

## Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit [www.PacktPub.com](http://www.PacktPub.com).

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

## Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

## Free access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.



# Preface

Natural Language Processing (NLP) has been used to address a wide range of problems, including support for search engines, summarizing and classifying text for web pages, and incorporating machine learning technologies to solve problems such as speech recognition and query analysis. It has found use wherever documents contain useful information.

NLP is used to enhance the utility and power of applications. It does so by making user input easier and converting text to more usable forms. In essence, NLP processes natural text found in a variety of sources, using a series of core NLP tasks to transform or extract information from the text.

This book focuses on core NLP tasks that will likely be encountered in an NLP application. Each NLP task presented in this book starts with a description of the problem and where it can be used. The issues that make each task difficult are introduced so that you can understand the problem in a better way. This is followed by the use of numerous Java techniques and APIs to support an NLP task.

## What this book covers

*Chapter 1, Introduction to NLP*, explains the importance and uses of NLP. The NLP techniques used in this chapter are explained with simple examples illustrating their use.

*Chapter 2, Finding Parts of Text*, focuses primarily on tokenization. This is the first step in more advanced NLP tasks. Both core Java and Java NLP tokenization APIs are illustrated.

*Chapter 3, Finding Sentences*, proves that sentence boundary disambiguation is an important NLP task. This step is a precursor for many other downstream NLP tasks where text elements should not be split across sentence boundaries. This includes ensuring that all phrases are in one sentence and supporting parts of speech analysis.

*Chapter 4, Finding People and Things*, covers what is commonly referred to as Named Entity Recognition. This task is concerned with identifying people, places, and similar entities in text. This technique is a preliminary step for processing queries and searches.

*Chapter 5, Detecting Parts of Speech*, shows you how to detect parts of speech, which are grammatical elements of text, such as nouns and verbs. Identifying these elements is a significant step in determining the meaning of text and detecting relationships within text.

*Chapter 6, Classifying Texts and Documents*, proves that classifying text is useful for tasks such as spam detection and sentiment analysis. The NLP techniques that support this process are investigated and illustrated.

*Chapter 7, Using Parser to Extract Relationships*, demonstrates parse trees. A parse tree is used for many purposes, including information extraction. It holds information regarding the relationships between these elements. An example implementing a simple query is presented to illustrate this process.

*Chapter 8, Combined Approaches*, contains techniques for extracting data from various types of documents, such as PDF and Word files. This is followed by an examination of how the previous NLP techniques can be combined into a pipeline to solve larger problems.

## What you need for this book

Java SDK 7 is used to illustrate the NLP techniques. Various NLP APIs are needed and can be readily downloaded. An IDE is not required but is desirable.

## Who this book is for

Experienced Java developers who are interested in NLP techniques will find this book useful. No prior exposure to NLP is required.

# Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and explanations of their meanings.

Code words in text are shown as follows: "The `keyset` method returns a set of all the annotation keys currently held by the `Annotation` object."

Database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "To demonstrate the use of POI, we will use a file called `TestDocument.pdf`."


A block of code is set as follows:


```
for (int index = 0; index < sentences.length; index++) {
    String tokens[] = tokenizer.tokenize(sentences[index]);
    Span nameSpans[] = nameFinder.find(tokens);
    for(Span span : nameSpans) {
        list.add("Sentence: " + index
            + " Span: " + span.toString() + " Entity: "
            + tokens[span.getStart()]);
    }
}
```

The output of code sequences looks like what is shown here:

```
Sentence: 0 Span: [0..1) person Entity: Joe
Sentence: 0 Span: [7..9) person Entity: Fred
Sentence: 2 Span: [0..1) person Entity: Joe
```

New terms and important words are shown in bold.

[  Warnings or important notes appear in a box like this. ]

[  Tips and tricks appear like this. ]

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book – what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

## **Piracy**

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

## **Questions**

If you have a problem with any aspect of this book, you can contact us at [questions@packtpub.com](mailto:questions@packtpub.com), and we will do our best to address the problem.

# Table of Contents

<b>Preface</b>	<b>vii</b>
<b>Chapter 1: Introduction to NLP</b>	<b>1</b>
What is NLP?	2
Why use NLP?	3
Why is NLP so hard?	4
Survey of NLP tools	6
Apache OpenNLP	7
Stanford NLP	8
LingPipe	10
GATE	11
UIMA	12
Overview of text processing tasks	12
Finding parts of text	13
Finding sentences	14
Finding people and things	16
Detecting Parts of Speech	18
Classifying text and documents	20
Extracting relationships	20
Using combined approaches	23
Understanding NLP models	23
Identifying the task	24
Selecting a model	24
Building and training the model	25
Verifying the model	25
Using the model	25
Preparing data	25
Summary	28

<b>Chapter 2: Finding Parts of Text</b>	<b>29</b>
<b>Understanding the parts of text</b>	<b>30</b>
<b>What is tokenization?</b>	<b>30</b>
Uses of tokenizers	32
<b>Simple Java tokenizers</b>	<b>33</b>
Using the Scanner class	33
Specifying the delimiter	34
Using the split method	35
Using the BreakIterator class	36
Using the StreamTokenizer class	37
Using the StringTokenizer class	39
Performance considerations with java core tokenization	40
<b>NLP tokenizer APIs</b>	<b>40</b>
Using the OpenNLPTokenizer class	41
Using the SimpleTokenizer class	41
Using the WhitespaceTokenizer class	42
Using the TokenizerME class	42
Using the Stanford tokenizer	43
Using the PTBTokenizer class	44
Using the DocumentPreprocessor class	45
Using a pipeline	46
Using LingPipe tokenizers	47
Training a tokenizer to find parts of text	48
Comparing tokenizers	52
<b>Understanding normalization</b>	<b>52</b>
Converting to lowercase	53
Removing stopwords	53
Creating a StopWords class	54
Using LingPipe to remove stopwords	56
Using stemming	57
Using the Porter Stemmer	58
Stemming with LingPipe	59
Using lemmatization	60
Using the StanfordLemmatizer class	60
Using lemmatization in OpenNLP	62
Normalizing using a pipeline	64
<b>Summary</b>	<b>65</b>
<b>Chapter 3: Finding Sentences</b>	<b>67</b>
<b>The SBD process</b>	<b>67</b>
<b>What makes SBD difficult?</b>	<b>68</b>
<b>Understanding SBD rules of LingPipe's</b>	
<b>HeuristicSentenceModel class</b>	<b>70</b>

---

<b>Simple Java SBDs</b>	<b>71</b>
Using regular expressions	71
Using the BreakIterator class	73
<b>Using NLP APIs</b>	<b>76</b>
Using OpenNLP	76
Using the SentenceDetectorME class	76
Using the sentPosDetect method	78
Using the Stanford API	79
Using the PTBTokenizer class	79
Using the DocumentPreprocessor class	83
Using the StanfordCoreNLP class	86
Using LingPipe	88
Using the IndoEuropeanSentenceModel class	88
Using the SentenceChunker class	90
Using the MedlineSentenceModel class	92
<b>Training a Sentence Detector model</b>	<b>93</b>
Using the Trained model	95
Evaluating the model using the SentenceDetectorEvaluator class	96
<b>Summary</b>	<b>97</b>
<b>Chapter 4: Finding People and Things</b>	<b>99</b>
<b>Why NER is difficult?</b>	<b>100</b>
<b>Techniques for name recognition</b>	<b>101</b>
Lists and regular expressions	101
Statistical classifiers	102
<b>Using regular expressions for NER</b>	<b>102</b>
Using Java's regular expressions to find entities	103
Using LingPipe's RegExChunker class	105
<b>Using NLP APIs</b>	<b>106</b>
Using OpenNLP for NER	107
Determining the accuracy of the entity	109
Using other entity types	110
Processing multiple entity types	111
Using the Stanford API for NER	113
Using LingPipe for NER	115
Using LingPipe's name entity models	115
Using the ExactDictionaryChunker class	117
<b>Training a model</b>	<b>119</b>
Evaluating a model	122
<b>Summary</b>	<b>123</b>



---

<b>Chapter 5: Detecting Parts of Speech</b>	<b>125</b>
<b>The tagging process</b>	<b>125</b>
Importance of POS taggers	128
What makes POS difficult?	128
<b>Using the NLP APIs</b>	<b>130</b>
Using OpenNLP POS taggers	131
Using the OpenNLP POSTaggerME class for POS taggers	132
Using OpenNLP chunking	134
Using the POSDictionary class	138
Using Stanford POS taggers	142
Using Stanford MaxentTagger	142
Using the MaxentTagger class to tag textese	145
Using Stanford pipeline to perform tagging	146
Using LingPipe POS taggers	149
Using the HmmDecoder class with Best_First tags	150
Using the HmmDecoder class with NBest tags	151
Determining tag confidence with the HmmDecoder class	152
Training the OpenNLP POSModel	154
<b>Summary</b>	<b>156</b>
<b>Chapter 6: Classifying Texts and Documents</b>	<b>157</b>
<b>How classification is used</b>	<b>157</b>
<b>Understanding sentiment analysis</b>	<b>159</b>
<b>Text classifying techniques</b>	<b>161</b>
<b>Using APIs to classify text</b>	<b>161</b>
Using OpenNLP	162
Training an OpenNLP classification model	162
Using DocumentCategorizerME to classify text	165
Using Stanford API	167
Using the ColumnDataClassifier class for classification	167
Using the Stanford pipeline to perform sentiment analysis	170
Using LingPipe to classify text	172
Training text using the Classified class	172
Using other training categories	174
Classifying text using LingPipe	175
Sentiment analysis using LingPipe	176
Language identification using LingPipe	178
<b>Summary</b>	<b>180</b>

---