自然语言标注——用于机器学习（影印版）

Natural Language Annotation

for Machine Learning

O'REILLY®
東南大學出版社

James Pustejovsky
& Amber Stubbs 著

自然语言标注——用于机器学习（影印版）

Natural Language Annotation for Machine Learning

James Pustejovsky & Amber Stubbs 著

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

O'Reilly Media, Inc.授权东南大学出版社出版

南京 东南大学出版社

图书在版编目（CIP）数据

自然语言标注：用于机器学习：英文/(美)普斯特若夫斯基（Pustejovsky, J.)著,(美)斯塔布斯（Stubbs, A.)著.—影印本.—南京：东南大学出版社，2013.6

书名原文：Natural Language Annotation for Machine Learning
ISBN 978-7-5641-4281-0

I.①自… II.①普… ②斯… III.①自然语言处理－英文
IV.①TP391

中国版本图书馆CIP数据核字（2013）第113429号

江苏省版权局著作权合同登记
图字：10-2013-132号

自然语言标注——用于机器学习（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
印　　刷：扬中市印刷有限公司
开　　本：787毫米×980毫米　16开本
印　　张：21.5
字　　数：421千字
版　　次：2013年6月第1版
印　　次：2013年6月第1次印刷
书　　号：ISBN 978-7-5641-4281-0
定　　价：54.00元（册）

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

Preface

This book is intended as a resource for people who are interested in using computers to help process natural language. A *natural language* refers to any language spoken by humans, either currently (e.g., English, Chinese, Spanish) or in the past (e.g., Latin, ancient Greek, Sanskrit). *Annotation* refers to the process of adding metadata information to the text in order to augment a computer's capability to perform Natural Language Processing (NLP). In particular, we examine how information can be added to natural language text through annotation in order to increase the performance of *machine learning algorithms*—computer programs designed to extrapolate rules from the information provided over texts in order to apply those rules to unannotated texts later on.

Natural Language Annotation for Machine Learning

This book details the multistage process for building your own annotated natural language dataset (known as a *corpus*) in order to train machine learning (ML) algorithms for language-based data and knowledge discovery. The overall goal of this book is to show readers how to create their own corpus, starting with selecting an annotation task, creating the annotation specification, designing the guidelines, creating a "gold standard" corpus, and then beginning the actual data creation with the annotation process.

Because the annotation process is not linear, multiple iterations can be required for defining the tasks, annotations, and evaluations, in order to achieve the best results for a particular goal. The process can be summed up in terms of the *MATTER Annotation Development Process*: Model, Annotate, Train, Test, Evaluate, Revise. This book guides the reader through the cycle, and provides detailed examples and discussion for different types of annotation tasks throughout. These tasks are examined in depth to provide context for readers and to help provide a foundation for their own ML goals.

Additionally, this book provides access to and usage guidelines for lightweight, user-friendly software that can be used for annotating texts and adjudicating the annotations. While a variety of annotation tools are available to the community, the Multipurpose Annotation Environment (MAE) adopted in this book (and available to readers as a free download) was specifically designed to be easy to set up and get running, so that confusing documentation would not distract readers from their goals. MAE is paired with the Multidocument Adjudication Interface (MAI), a tool that allows for quick comparison of annotated documents.

Audience

This book is written for anyone interested in using computers to explore aspects of the information content conveyed by natural language. It is not necessary to have a programming or linguistics background to use this book, although a basic understanding of a scripting language such as Python can make the MATTER cycle easier to follow, and some sample Python code is provided in the book. If you don't have any Python experience, we highly recommend *Natural Language Processing with Python* by Steven Bird, Ewan Klein, and Edward Loper (O'Reilly), which provides an excellent introduction both to Pvthon and to aspects of NLP that are not addressed in this book.

It is helpful to have a basic understanding of markup languages such as XML (or even HTML) in order to get the most out of this book. While one doesn't need to be an expert in the theory behind an XML schema, most annotation projects use some form of XML to encode the tags, and therefore we use that standard in this book when providing annotation examples. Although you don't need to be a web designer to understand the book, it does help to have a working knowledge of tags and attributes in order to understand how an idea for an annotation gets implemented.

Organization of This Book

Chapter 1 of this book provides a brief overview of the history of annotation and machine learning, as well as short discussions of some of the different ways that annotation tasks have been used to investigate different layers of linguistic research. The rest of the book guides the reader through the MATTER cycle, from tips on creating a reasonable annotation goal in Chapter 2, all the way through evaluating the results of the annotation and ML stages, as well as a discussion of revising your project and reporting on your work in Chapter 9. The last two chapters give a complete walkthrough of a single annotation project and how it was recreated with machine learning and rule-based algorithms. Appendixes at the back of the book provide lists of resources that readers will find useful for their own annotation tasks.

Software Requirements

While it's possible to work through this book without running any of the code examples provided, we do recommend having at least the Natural Language Toolkit (NLTK) installed for easy reference to some of the ML techniques discussed. The NLTK currently runs on Python versions from 2.4 to 2.7. (Python 3.0 is not supported at the time of this writing.) For more information, see *http://www.nltk.org*.

The code examples in this book are written as though they are in the interactive Python shell programming environment. For information on how to use this environment, please see: *http://docs.python.org/tutorial/interpreter.html*. If not specifically stated in the examples, it should be assumed that the command `import nltk` was used prior to all sample code.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

This icon signifies a tip, suggestion, or general note.

This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Natural Language Annotation for Machine Learning* by James Pustejovsky and Amber Stubbs (O'Reilly). Copyright 2013 James Pustejovsky and Amber Stubbs, 978-1-449-30666-3."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online (*www.safaribooksonline.com*) is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://oreil.ly/nat-lang-annotation-ML*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com.*

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com.*

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

We would like thank everyone at O'Reilly who helped us create this book, in particular Meghan Blanchette, Julie Steele, Sarah Schneider, Kristen Borg, Audrey Doyle, and everyone else who helped to guide us through the process of producing it. We would also like to thank the students who participated in the Brandeis COSI 216 class during the spring 2011 semester for bearing with us as we worked through the MATTER cycle with them: Karina Baeza Grossmann-Siegert, Elizabeth Baran, Bensiin Borukhov, Nicholas Botchan, Richard Brutti, Olga Cherenina, Russell Entrikin, Livnat Herzig, Sophie Kushkuley, Theodore Margolis, Alexandra Nunes, Lin Pan, Batia Snir, John Vogel, and Yaqin Yang.

We would also like to thank our technical reviewers, who provided us with such excellent feedback: Arvind S. Gautam, Catherine Havasi, Anna Rumshisky, and Ben Wellner, as well as everyone who read the Early Release version of the book and let us know that we were going in the right direction.

We would like to thank members of the ISO community with whom we have discussed portions of the material in this book: Kiyong Lee, Harry Bunt, Nancy Ide, Nicoletta Calzolari, Bran Boguraev, Annie Zaenen, and Laurent Romary.

Additional thanks to the members of the Brandeis Computer Science and Linguistics departments, who listened to us brainstorm, kept us encouraged, and made sure everything kept running while we were writing, especially Marc Verhagen, Lotus Goldberg, Jessica Moszkowicz, and Alex Plotnick.

This book could not exist without everyone in the linguistics and computational linguistics communities who have created corpora and annotations, and, more importantly, shared their experiences with the rest of the research community.

James Adds:

I would like to thank my wife, Cathie, for her patience and support during this project. I would also like to thank my children, Zac and Sophie, for putting up with me while the book was being finished. And thanks, Amber, for taking on this crazy effort with me.

Amber Adds:

I would like to thank my husband, BJ, for encouraging me to undertake this project and for his patience while I worked through it. Thanks also to my family, especially my parents, for their enthusiasm toward this book. And, of course, thanks to my advisor and coauthor, James, for having this crazy idea in the first place.

Table of Contents

CHAPTER 1
The Basics

It seems as though every day there are new and exciting problems that people have taught computers to solve, from how to win at chess or *Jeopardy* to determining shortest-path driving directions. But there are still many tasks that computers cannot perform, particularly in the realm of understanding human language. Statistical methods have proven to be an effective way to approach these problems, but machine learning (ML) techniques often work better when the algorithms are provided with pointers to what is relevant about a dataset, rather than just massive amounts of data. When discussing natural language, these pointers often come in the form of annotations—metadata that provides additional information about the text. However, in order to teach a computer effectively, it's important to give it the right data, and for it to have enough data to learn from. The purpose of this book is to provide you with the tools to create good data for your own ML task. In this chapter we will cover:

- Why annotation is an important tool for linguists and computer scientists alike
- How corpus linguistics became the field that it is today
- The different areas of linguistics and how they relate to annotation and ML tasks
- What a corpus is, and what makes a corpus balanced
- How some classic ML problems are represented with annotations
- The basics of the annotation development cycle

The Importance of Language Annotation

Everyone knows that the Internet is an amazing resource for all sorts of information that can teach you just about anything: juggling, programming, playing an instrument, and so on. However, there is another layer of information that the Internet contains, and that is how all those lessons (and blogs, forums, tweets, etc.) are being communicated. The Web contains information in all forms of media—including texts, images, movies, and sounds—and language is the communication medium that allows people to understand the content, and to link the content to other media. However, while computers are excellent at delivering this information to interested users, they are much less adept at understanding language itself.

Theoretical and computational linguistics are focused on unraveling the deeper nature of language and capturing the computational properties of linguistic structures. Human language technologies (HLTs) attempt to adopt these insights and algorithms and turn them into functioning, high-performance programs that can impact the ways we interact with computers using language. With more and more people using the Internet every day, the amount of linguistic data available to researchers has increased significantly, allowing linguistic modeling problems to be viewed as ML tasks, rather than limited to the relatively small amounts of data that humans are able to process on their own.

However, it is not enough to simply provide a computer with a large amount of data and expect it to learn to speak—the data has to be prepared in such a way that the computer can more easily find patterns and inferences. This is usually done by adding relevant metadata to a dataset. Any metadata tag used to mark up elements of the dataset is called an *annotation* over the input. However, in order for the algorithms to learn efficiently and effectively, the annotation done on the data must be accurate, and relevant to the task the machine is being asked to perform. For this reason, the discipline of language annotation is a critical link in developing intelligent human language technologies.

Giving an ML algorithm too much information can slow it down and lead to inaccurate results, or result in the algorithm being so molded to the training data that it becomes “overfit” and provides less accurate results than it might otherwise on new data. It’s important to think carefully about what you are trying to accomplish, and what information is most relevant to that goal. Later in the book we will give examples of how to find that information, and how to determine how well your algorithm is performing at the task you’ve set for it.

Datasets of natural language are referred to as *corpora*, and a single set of data annotated with the same specification is called an *annotated corpus*. Annotated corpora can be used to train ML algorithms. In this chapter we will define what a corpus is, explain what is meant by an annotation, and describe the methodology used for enriching a linguistic data collection with annotations for machine learning.