Lang Resources & Evaluation (2010) 44:421–424
DOI 10.1007/s10579-010-9124-x

Steven Bird, Ewan Klein and Edward Loper: Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit

O'Reilly Media, Beijing, 2009, ISBN 978-0-596-51649-9

Wiebke Wagner

Published online: 27 May 2010

Natural Language Processing (NLP) is experiencing rapid growth as its theories and methods are more and more deployed in a wide range of different fields. In the humanities, the work on corpora is gaining increasing prominence. Within industry, people need NLP for market analysis, web software development to name a few examples. For this reason it is important for many people to have some working knowledge of NLP. The book "Natural Language Processing with Python" by Steven Bird, Ewan Klein and Edward Loper is a recent contribution to cover this demand. It introduces the freely available Natural Language Toolkit (NLTK)[1]—a project by the same authors—that was designed with the following goals: simplicity, consistency, extensibility and modularity.

The book pursues pedagogical aims and is intended for students or others who want to learn to write programs that analyze natural language. Programming knowledge is not necessarily expected since the book is written for people "new to programming", "new to Python" and "already dreaming in Python" (p. x). Furthermore it targets lecturers who can use it in their courses.

The book is a practical guide to NLP, achieving a balance between NLP theory and practical programming skills. It alternates between focusing on natural language, supported by pertinent programming examples, or focusing on the Python programming language while linguistic examples play a supporting role. The reader gets to know many real-world NLP applications and learns by example.

The book is well structured. Each chapter starts with some key questions that give a rough idea what information will be provided in the chapter. The chapters finish with a summary, exercises of levels "easy", "intermediate" and "difficult"

[1] http://www.nltk.org.

W. Wagner (✉)
Institut for Natural Language Processing, University of Stuttgart,
70127 Stuttgart, Germany
e-mail: wiebke.wagner@ims.uni-stuttgart.de

and finally with a further-reading section. The latter contains carefully selected literature, URLs to lexical resources and relevant communities.

The early chapters up to Chap. 4 lay the programming foundations needed in the book and introduce rather simple NLP applications. Chapter 1 dives straight into Python programming, introducing conditions, loops and list comprehensions as well as different data types like strings, lists and sets. The introduction to the programming language is illustrated in simple NLP applications like searching, counting, computing simple statistics like word frequency distributions etc. At this point, the reader is already able to write quite a few NLP applications which is motivating and fun. Chapter 2 discusses different corpora and lexical resources. Many nice examples show what can be done with large amounts of data, e.g. exploring the evolving usage of the words 'America' and 'citizen' over a range of time. Chapter 3 again displays a lot of basic Python features including files, encodings and regular expressions combined with the NLP topics of tokenization, lemmatization, stemming as well as word and sentence segmentation. Chapter 4 follows a more conventional approach to teaching programming and describes systematically the most important Python concepts that have not already been introduced. Later on in the chapter, the authors give insight into more sophisticated programming features like accumulative functions, named arguments and techniques for designing algorithms like dynamic programming.

In the following Chapters fundamental NLP topics are covered. Chapter 5 concerns part-of-speech tagging following both rule based and statistical approaches. In Chap. 6 the reader becomes acquainted with supervised classification. They are equipped with techniques to extract features, to split data into training, test and development-test sets, to train classifiers and to compute accuracies. Explicative examples show the usage of Decision Trees, Naive Bayes classifiers and Maximum-Entropy classifiers. The focus of Chap. 7 is information extraction. Here chunking is introduced, which is used for named entity recognition and relation extraction. Chapters 8 and 9 are concerned with the topic of parsing. Chapter 8 focuses on the representation of sentence structures, on parsers and on grammar development. Whereas Chap. 9 shows implementations of fine grained feature structures that cover a variety of linguistic phenomena. Chapter 10, "Analyzing the meaning of sentences", turns to logic, providing an overview of propositional logic, predicate logic, model building and compositionality. To assign an English sentence its truth conditions, the grammar framework from Chap. 9 is used to parse the sentence into a logical form. The Chapter finishes with a section about discourse semantics. The final chapter, "Managing linguistic data" discusses structures and sources of corpora, annotation levels and quality control, exemplified by the TIMIT corpus. The book closes with an Afterword where the authors give a brief summary of the history of the field, discussing symbolic versus statistical approaches and the background behind it. An NLTK Roadmap of further NLTK developments points out areas missing in the toolkit including lexical semantics, language generation, phonology and morphology among others and encourages the reader to contribute new NLP implementations to NLTK.

Throughout the book, the authors have to find the right balance between what the reader has to know and what would be too much detail; how much programming

knowledge is essential and how much theoretical background is needed to apply a certain NLP method – a subtle problem the authors manage to balance very well.

The book introduces programming in quite an unusual order, beginning with non-trivial data types and non-trivial control structures. This approach gives the reader the ability to do useful language processing from the start. All relevant Python features are carefully explained and exemplified in small scripts. However, the reader without any programming knowledge will probably find it quite difficult to fully understand the Python syntax, e.g. when to use which kind of brackets , why to type `w.upper()` and not `upper(w)` but `len(w)`. Readers completely unfamiliar with programming probably need to have a look at other references like Lutz (2009) or those given in the Further Reading section of Chap. 4.

An other unusual thing about order is that concepts are sometimes applied before they are explained. The NLTK method `ibigrams()` suddenly appears in a piece of code (p. 187) or in Chap. 6 the decision tree classifier is used before it is made clear what a decision tree classifier is. However, trust on the authors; usually they catch up a bit later. The more curious readers can look it up in the index of the book.

The book is very strong on code. Many detailed and entertaining examples show NLP applications in a cookbook-like style. The code is clean and clear, and the toolkit is easy to install. If something does not work, precise instructions are given. However, NLTK is not optimized for runtime performance since it is all written in Python. For researchers working with mounds of data or with computationally intensive processes, the NLTK software is probably too slow. Still it is useful for first experiments or for smaller ad-hoc tasks: traversing WordNet to retrieve the distance or relation of two entities, annotating or parsing some data, proving a logical theorem and many more. In addition: if the book is used in a course for teaching, speed will not be the central point.

One of the strengths of the book is that the authors really take time to explain how NLP problems arise and why sophisticated techniques are necessary to solve them. So, they show on several pages why pp-attachment resolution is such a crucial task (Chap. 8, p. 298ff) or why databases are not sufficient for the representation of meaning (Chap. 10, p. 361ff).

Since the focus of the book lies mainly on practical solutions for language processing, the authors cannot provide very deep insights into theoretical issues. Especially Chaps. 8 and 10 suffer somewhat from the balancing act between theory and practical application. In these chapters, the underlying theory gets more important and at the same time very complex. Here, a reader with some theoretical background knowledge is at an advantage, and they will probably get more out of the later chapters in the book. But again, the further-reading sections point to relevant references, since there are quite a few NLP introductions on the market like Jurafsky and Martin (2009).

To summarize: The maxim of the book is: learning by doing. The book introduces quite a bit of Python programming, some NLP theory and a lot of how to solve real NLP problems. The goal to take very beginners in either fields through the book, might be a bit too ambitious. Still, the practical approach by means of the NLTK framework makes it very valuable and—as far as the reviewer is concerned—unique in the scope of NLP literature.

The book is freely available on the NLTK homepage.[2] So, have a look at it and—to cite the last words of the book—"happy hacking!".

References

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing, speech recognition, and computational linguistics* (2nd ed.). New Jersey: Prentice-Hall.

Lutz, M. (2009). *Learning python* (4th ed.). Beijing: O'Reilly Media.

[2] http://www.nltk.org/book.