Preface

Most human knowledge — and most human communication — is represented and expressed using language. Language technologies permit computers to process human language automatically; handheld computers support predictive text and handwriting recognition; web search engines give access to information locked up in unstructured text. By providing more natural human-machine interfaces, and more sophisticated access to stored information, language processing has come to play a central role in the multilingual information society.

This textbook provides a comprehensive introduction to the field of natural language processing (NLP), covering the major techniques and theories. The book provides numerous worked examples and exercises, and can serve as the main text for undergraduate and introductory graduate courses on natural language processing or computational linguistics.

Audience

This book is intended for people in the language sciences and the information sciences who want to learn how to write programs that analyze written language. You won't need any prior knowledge of linguistics or computer science; those with a background in either area can simply skip over some of the discussions. Depending on which background you come from, and your motivation for being interested in NLP, you will gain different kinds of skills and knowledge from this book, as set out below:

Goals	Background	
	Linguistics	Computer Science
Linguistic Analysis	Programming to manage linguistic data, explore formal theories, and test empirical claims	Linguistics as a source of interesting problems in data modelling, data mining, and formal language theory
Language Technology	Learning to program with applications to familiar problems, to work in language technology or other technical field	Knowledge of linguistics as fundamental to developing high quality, maintainable language processing software

Table 1:

The structure of the book is biased towards linguists, in that the introduction to programming appears in the main chapter sequence, and early chapters contain many elementary examples. We hope that computer science readers can quickly skip over such materials till they reach content that is more linguistically challenging.

What You Will Learn

By the time you have dug into the material presented here, you will have acquired substantial skills and knowledge in the following areas:

- how simple programs can help linguists manipulate and analyze language data, and how to write these programs;
- key concepts from linguistic description and analysis;
- how linguistic knowledge is used in important language technology components;
- knowledge of the principal data structures and algorithms used in NLP, and skills in algorithmic problem solving, data modelling, and data management;
- understanding of the standard corpora and their use in formal evaluation;
- the organization of the field of NLP;
- skills in Python programming for NLP.

Download the Toolkit...

This textbook is a companion to the *Natural Language Toolkit*. All software, corpora, and documentation are freely downloadable from http://nltk.sourceforge.net/. Distributions are provided for Windows, Macintosh and Unix platforms. All NLTK distributions plus Python and WordNet distributions are also available in the form of an ISO image which can be downloaded and burnt to CD-ROM for easy local redistribution. We strongly encourage you to download the toolkit before you go beyond the first chapter of the book.

Emphasis

This book is a **practical** introduction to NLP. You will learn by example, write real programs, and grasp the value of being able to test an idea through implementation. If you haven't learnt already, this book will teach you **programming**. Unlike other programming books, we provide extensive illustrations and exercises from NLP. The approach we have taken is also **principled**, in that we cover the theoretical underpinnings and don't shy away from careful linguistic and computational analysis. We have tried to be **pragmatic** in striking a balance between theory and application, and alternate between the two several times each chapter, identifying the connections but also the tensions. Finally, we recognize that you won't get through this unless it is also **pleasurable**, so we have tried to include many applications and examples that are interesting and entertaining, sometimes whimsical.

Structure

The book is structured into three parts, as follows:

Part 1: Basics In this part, we focus on recognising simple structure in text. We start with individual words, then explore parts of speech and simple syntactic constituents.

Part 2: Parsing Here, we deal with syntactic structure, trees, grammars, and parsing.

Part 3: Advanced Topics This final part of the book contains chapters which address selected topics in NLP in more depth and to a more advanced level. By design, the chapters in this part can be read independently of each other.

The three parts have a common structure: they start off with a chapter on programming, followed by three chapters on various topics in NLP. The programming chapters are *foundational*, and you must master this material before progressing further.

Each chapter consists of an introduction, a sequence of sections that progress from elementary to advanced material, and finally a summary and suggestions for further reading. Most sections include exercises which are graded according to the following scheme: ☼ is for easy exercises that involve minor modifications to supplied code samples or other simple activities; ◑ is for intermediate exercises that explore an aspect of the material in more depth, requiring careful analysis and design; ★ is for difficult, open-ended tasks that will challenge your understanding of the material and force you to think independently (readers new to programming are encouraged to skip these). The exercises are important for consolidating the material in each section, and we strongly encourage you to try a few before continuing with the rest of the chapter.

For Instructors

Natural Language Processing (NLP) is often taught within the confines of a single-semester course at advanced undergraduate level or postgraduate level. Many instructors have found that it is difficult to cover both the theoretical and practical sides of the subject in such a short span of time. Some courses focus on theory to the exclusion of practical exercises, and deprive students of the challenge and excitement of writing programs to automatically process language. Other courses are simply designed to teach programming for linguists, and do not manage to cover any significant NLP content. The *Natural Language Toolkit* (NLTK) was developed to address this problem, making it feasible to cover a substantial amount of theory and practice within a single-semester course, even if students have no prior programming experience.

A significant fraction of any NLP syllabus covers fundamental data structures and algorithms. These are usually taught with the help of formal notations and complex diagrams. Large trees and charts are copied onto the board and edited in tedious slow motion, or laboriously prepared for presentation slides. It is more effective to use live demonstrations in which those diagrams are generated and updated automatically. NLTK provides interactive graphical user interfaces, making it possible to view program state and to study program execution step-by-step. Most NLTK components have a demonstration mode, and will perform an interesting task without requiring any special input from the user. It is even possible to make minor modifications to programs in response to "what if" questions. In this way, students learn the mechanics of NLP quickly, gain deeper insights into the data structures and algorithms, and acquire new problem-solving skills.

NLTK supports assignments of varying difficulty and scope. In the simplest assignments, students experiment with existing components to perform a wide variety of NLP tasks. This may involve no programming at all, in the case of the existing demonstrations, or simply changing a line or two of program code. As students become more familiar with the toolkit they can be asked to modify existing components or to create complete systems out of existing components. NLTK also provides students with a flexible framework for advanced projects, such as developing a multi-component system, by integrating and extending NLTK components, and adding on entirely new components. Here NLTK

helps by providing standard implementations of all the basic data structures and algorithms, interfaces to standard corpora, substantial corpus samples, and a flexible and extensible architecture. Thus, as we have seen, NLTK offers a fresh approach to NLP pedagogy, in which theoretical content is tightly integrated with application.

We believe our book is unique in providing a comprehensive pedagogical framework for students to learn about NLP in the context of learning to program. What sets our materials apart is the tight coupling of the chapters and exercises with NLTK, giving students — even those with no prior programming experience — a practical introduction to NLP. Once completing these materials, students will be ready to attempt one of the more advanced textbooks, such as *Foundations of Statistical Natural Language Processing*, by Manning and Schütze (MIT Press, 2000).

Course Plans; Lectures/Lab Sessions per Chapter		
Chapter	**Linguists**	**Computer Scientists**
1 Introduction	1	1
2 Programming	4	1
3 Words	2	2
4 Tagging	2-3	2
5 Chunking	0-2	2
6 Structured Programming	2-4	1
7 Grammars and Parsing	2-4	2-4
8 Advanced Parsing	1-4	3
9 Feature Based Grammar	2-4	2-4
10-14 Advanced Topics	2-8	2-16
Total	18-36	18-36

Table 2:

Further Reading:

The Association for Computational Linguistics (ACL) The ACL is the foremost professional body in NLP. Its journal and conference proceedings, approximately 10,000 articles, are available online with a full-text search interface, via http://www.aclweb.org/anthology/.

Linguistic Terminology A comprehensive glossary of linguistic terminology is available at:

http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/.

Language Files *Materials for an Introduction to Language and Linguistics (Ninth Edition)*, The Ohio State University Department of Linguistics. For more information, see http://www.ling.ohio-state.edu/publications/files/.

Acknowledgements

NLTK was originally created as part of a computational linguistics course in the Department of Computer and Information Science at the University of Pennsylvania in 2001. Since then it has been developed and expanded with the help of dozens of contributors. It has now been adopted in courses in dozens of universities, and serves as the basis of many research projects.

In particular, we're grateful to the following people for their feedback, comments on earlier drafts, advice, contributions: Greg Aumann, Ondrej Bojar, Trevor Cohn, James Curran, Jean Mark Gawron, Baden Hughes, Christopher Maloof, Stuart Robinson, Rob Speer. Many others have contributed to the toolkit, and they are listed at http://nltk.sourceforge.net/contrib.html.

We also acknowledge the following sources: Carpenter and Chu-Carroll's ACL-99 Tutorial on Spoken Dialogue Systems (example dialogue in 1).

About the Authors

		
Steven Bird	Ewan Klein	Edward Loper

Table 3:

Steven Bird is an Associate Professor in the Department of Computer Science and Software Engineering at the University of Melbourne, and a Senior Research Associate in the Linguistic Data Consortium at the University of Pennsylvania. After completing a PhD at the University of Edinburgh on computational phonology (1990), Steven moved to Cameroon to conduct fieldwork on tone and orthography. Later he spent four years as Associate Director of the Linguistic Data Consortium where he developed models and tools for linguistic annotation. His current research interests are in linguistic databases and query languages.

Ewan Klein is Professor of Language Technology in the School of Informatics at the University of Edinburgh. He completed a PhD on formal semantics at the University of Cambridge in 1978. After some years working at the Universities of Sussex and Newcastle upon Tyne, he took up a teaching position at Edinburgh. His current research interests are in computational semantics.

Edward Loper is a doctoral student in the Department of Computer and Information Sciences at the University of Pennsylvania, conducting research on machine learning in NLP. In addition to NLTK, he has helped develop other major packages for documenting and testing Python software, epydoc and doctest.

About this document...

This chapter is a draft from *Introduction to Natural Language Processing*, by Steven Bird, Ewan Klein and Edward Loper, Copyright © 2007 the authors. It is distributed with the *Natural Language Toolkit* [http://nltk.sourceforge.net], Version 0.7.5, under the terms of the *Creative Commons Attribution-ShareAlike License* [http://creativecommons.org/licenses/by-sa/2.5/].

This document is Revision: 4518 Wed May 16 20:08:28 EST 2007