Afterword: The Language Challenge

Natural language throws up some interesting computational challenges. We've explored many of these in the preceding chapters, including tokenization, tagging, classification, information extraction, and building syntactic and semantic representations. You should now be equipped to work with large datasets, to create robust models of linguistic phenomena, and to extend them into components for practical language technologies. We hope that the Natural Language Toolkit (NLTK) has served to open up the exciting endeavor of practical natural language processing to a broader audience than before.

In spite of all that has come before, language presents us with far more than a temporary challenge for computation. Consider the following sentences which attest to the riches of language:

(1)
a. Overhead the day drives level and grey, hiding the sun by a flight of grey spears. (William Faulkner, *As I Lay Dying*, 1935)

b. When using the toaster please ensure that the exhaust fan is turned on. (sign in dormitory kitchen)

c. Amiodarone weakly inhibited CYP2C9, CYP2D6, and CYP3A4-mediated activities with Ki values of 45.1-271.6 μM (Medline, PMID: 10718780)

d. Iraqi Head Seeks Arms (spoof news headline)

e. The earnest prayer of a righteous man has great power and wonderful results. (James 5:16b)

f. Twas brillig, and the slithy toves did gyre and gimble in the wabe (Lewis Carroll, *Jabberwocky*, 1872)

g. There are two ways to do this, AFAIK :smile: (internet discussion archive)

Other evidence for the riches of language is the vast array of disciplines whose work centers on language. Some obvious disciplines include translation, literary criticism, philosophy, anthropology and psychology. Many less obvious disciplines investigate language use, including law, hermeneutics, forensics, telephony, pedagogy, archaeology, cryptanalysis and speech pathology. Each applies distinct methodologies to gather observations, develop theories and test hypotheses. All serve to deepen our understanding of language and of the intellect that is manifested in language.

In view of the complexity of language and the broad range of interest in studying it from different angles, it's clear that we have barely scratched the surface here. Additionally, within NLP itself, there are many important methods and applications that we haven't mentioned.

In our closing remarks we will take a broader view of NLP, including its foundations and the further directions you might want to explore. Some of the topics are not well-supported by NLTK, and you might like to rectify that problem by contributing new software and data to the toolkit.

Language Processing vs Symbol Processing

The very notion that natural language could be treated in a computational manner grew out of a research program, dating back to the early 1900s, to reconstruct mathematical reasoning using logic, most clearly manifested in work by Frege, Russell, Wittgenstein, Tarski, Lambek and Carnap. This work led to the notion of language as a formal system amenable to automatic processing. Three later developments laid the foundation for natural language processing. The first was **formal language theory**. This defined a language as a set of strings accepted by a class of automata, such as context-free languages and pushdown automata, and provided the underpinnings for computational syntax.

The second development was **symbolic logic**. This provided a formal method for capturing selected aspects of natural language that are relevant for expressing logical proofs. A formal calculus in symbolic logic provides the syntax of a language, together with rules of inference and, possibly, rules of interpretation in a set-theoretic model; examples are propositional logic and First Order Logic. Given such a calculus, with a well-defined syntax and semantics, it becomes possible to associate meanings with expressions of natural language by translating them into expressions of the formal calculus. For example, if we translate *John saw Mary* into a formula `saw(j,m)`, we (implicitly or explicitly) intepret the English verb *saw* as a binary relation, and *John* and *Mary* as denoting individuals. More general statements like *All birds fly* require quantifiers, in this case ∀, meaning *for all*: ∀*x (bird(x) → fly(x))*. This use of logic provided the technical machinery to perform inferences that are an important part of language understanding.

A closely related development was the **principle of compositionality**, namely that the meaning of a complex expression is composed from the meaning of its parts and their mode of combination (10.). This principle provided a useful correspondence between syntax and semantics, namely that the meaning of a complex expression could be computed recursively. Consider the sentence *It is not true that p*, where *p* is a proposition. We can represent the meaning of this sentence as *not(p)*. Similarly, we can represent the meaning of *John saw Mary* as *saw(j, m)*. Now we can compute the interpretation of *It is not true that John saw Mary* recursively, using the above information, to get *not(saw(j,m))*.

The approaches just outlined share the premise that computing with natural language crucially relies on rules for manipulating symbolic representations. For a certain period in the development of NLP, particularly during the 1980s, this premise provided a common starting point for both linguists and practitioners of NLP, leading to a family of grammar formalisms known as unification-based (or feature-based) grammar (cf. 9.), and to NLP applications implemented in the Prolog programming language. Although grammar-based NLP is still a significant area of research, it has become somewhat eclipsed in the last 15–20 years due to a variety of factors. One significant influence came from automatic speech recognition. Although early work in speech processing adopted a model that emulated the kind of rule-based phonological processing typified by the *Sound Pattern of English* (Chomsky & Halle, 1968), this turned out to be hopelessly inadequate in dealing with the hard problem of recognizing actual speech in anything like real time. By contrast, systems which involved learning patterns from large bodies of speech data were significantly more accurate, efficient and robust. In addition, the speech community found that progress in building better systems was hugely assisted by the construction of shared resources for quantitatively measuring performance against common test data. Eventually, much of the NLP community embraced a **data intensive** orientation to language processing, coupled with a growing use of machine-learning techniques and evaluation-led methodology.

Contemporary Philosophical Divides

The contrasting approaches to NLP described in the preceding section relate back to early metaphysical debates about **rationalism** versus **empiricism** and **realism** versus **idealism** that occurred in the Enlightenment period of Western philosophy. These debates took place against a backdrop of orthodox thinking in which the source of all knowledge was believed to be divine revelation. During this period of the seventeenth and eighteenth centuries, philosophers argued that human reason or sensory experience has priority over revelation. Descartes and Leibniz, amongst others, took the rationalist position, asserting that all truth has its origins in human thought, and in the existence of "innate ideas" implanted in our minds from birth. For example, they argued that the principles of Euclidean geometry were developed using human reason, and were not the result of supernatural revelation or sensory experience. In contrast, Locke and others took the empiricist view, that our primary source of knowledge is the experience of our faculties, and that human reason plays a secondary role in reflecting on that experience. Often-cited evidence for this position was Galileo's discovery — based on careful observation of the motion of the planets — that the solar system is heliocentric and not geocentric. In the context of linguistics, this debate leads to the following question: to what extent does human linguistic experience, versus our innate "language faculty", provide the basis for our knowledge of language? In NLP this issue surfaces in debates about the priority of corpus data versus linguistic introspection in the construction of computational models.

A further concern, enshrined in the debate between realism and idealism, was the metaphysical status of the constructs of a theory. Kant argued for a distinction between phenomena, the manifestations we can experience, and "things in themselves" which can never be known directly. A linguistic realist would take a theoretical construct like **noun phrase** to be a real world entity that exists independently of human perception and reason, and which actually *causes* the observed linguistic phenomena. A linguistic idealist, on the other hand, would argue that noun phrases, along with more abstract constructs like semantic representations, are intrinsically unobservable, and simply play the role of useful fictions. The way linguists write about theories often betrays a realist position, while NLP practitioners occupy neutral territory or else lean towards the idealist position. Thus, in NLP, it is often enough if a theoretical abstraction leads to a useful result; it does not matter whether this result sheds any light on human linguistic processing.

These issues are still alive today, and show up in the distinctions between symbolic vs statistical methods, deep vs shallow processing, binary vs gradient classifications, and scientific vs engineering goals. However, such contrasts are now highly nuanced, and the debate is no longer as polarized as it once was. In fact, most of the discussions — and most of the advances even — involve a "balancing act." For example, one intermediate position is to assume that humans are innately endowed with analogical and memory-based learning methods (weak rationalism), and to use these methods to identify meaningful patterns in their sensory language experience (empiricism).

We have seen many examples of this methodology throughout this book. Statistical methods inform symbolic models any time corpus statistics guide the selection of productions in a context free grammar, i.e. "grammar engineering." Symbolic methods inform statistical models any time a corpus that was created using rule-based methods is used as a source of features for training a statistical language model, i.e. "grammatical inference." The circle is closed.

NLTK Roadmap

The Natural Language Toolkit is work-in-progress, and is being continually expanded as people contribute code. Some areas of NLP and linguistics are not (yet) well supported in NLTK, and contributions in these areas are especially welcome. Check `http://nltk.org/` for news about developments after the publication date of the book.

Phonology and Morphology:
Computational approaches to the study of sound patterns and word structures typically use a finite state toolkit. Phenomena such as suppletion and non-concatenative morphology are difficult to address using the string processing methods we have been studying. The technical challenge is not only to link NLTK to a high-performance finite state toolkit, but to avoid duplication of lexical data and to link the morphosyntactic features needed by morph analyzers and syntactic parsers.

High-Performance Components:
Some NLP tasks are too computationally intensive for pure Python implementations to be feasible. However, in some cases the expense only arises when training models, not when using them to label inputs. NLTK's package system provides a convenient way to distribute trained models, even models trained using corpora that cannot be freely distributed. Alternatives are to develop Python interfaces to high-performance machine learning tools, or to expand the reach of Python by using parallel programming techniques like MapReduce.

Lexical Semantics:
This is a vibrant area of current research, encompassing inheritance models of the lexicon, ontologies, multiword expressions, etc, mostly outside the scope of NLTK as it stands. A conservative goal would be to access lexical information from rich external stores in support of tasks in word sense disambiguation, parsing, and semantic interpretation.

Natural Language Generation:
Producing coherent text from underlying representations of meaning is an important part of NLP; a unification based approach to NLG has been developed in NLTK, and there is scope for more contributions in this area.

Linguistic Fieldwork:
A major challenge faced by linguists is to document thousands of endangered languages, work which generates heterogeneous and rapidly evolving data in large quantities. More fieldwork data formats, including interlinear text formats and lexicon interchange formats, could be supported in NLTK, helping linguists to curate and analyze this data, while liberating them to spend as much time as possible on data elicitation.

Other Languages:
Improved support for NLP in languages other than English could involve work in two areas: obtaining permission to distribute more corpora with NLTK's data collection; writing language-specific HOWTOs for posting at `http://nltk.org/howto`, illustrating the use of NLTK and discussing language-specific problems for NLP including character encodings, word segmentation, and morphology. NLP researchers with expertise in a particular language could arrange to translate this book and host a copy on the NLTK website; this would go beyond translating the discussions to providing equivalent worked examples using data in the target language, a non-trivial undertaking.

NLTK-Contrib: Many of NLTK's core components were contributed by members of the NLP community, and were initially housed in NLTK's "Contrib" package, `nltk_contrib`. The only requirement for software to be added to this package is that it must be written in Python, relevant to NLP, and given the same open source license as the rest of NLTK. Imperfect software is welcome, and will probably be improved over time by other members of the NLP community.

Teaching Materials:
Since the earliest days of NLTK development, teaching materials have accompanied the software, materials that have gradually expanded to fill this book, plus a substantial quantity of online materials as well. We hope that instructors who supplement these materials with presentation slides, problem sets, solution sets, and more detailed treatments of the topics we have covered, will make them available, and will notify the authors so we can link them from `http://nltk.org/`. Of particular value are materials that help NLP become a mainstream

	course in the undergraduate programs of computer science and linguistics departments, or that make NLP accessible at the secondary level where there is significant scope for including computational content in the language, literature, computer science, and information technology curricula.
Only a Toolkit:	As stated in the preface, NLTK is a *toolkit*, not a system. Many problems will be tackled with a combination of NLTK, Python, other Python libraries, and interfaces to external NLP tools and formats.

Envoi...

Linguists are sometimes asked how many languages they speak, and have to explain that this field actually concerns the study of abstract structures that are shared by languages, a study which is more profound and elusive than learning to speak as many languages as possible. Similarly, computer scientists are sometimes asked how many programming languages they know, and have to explain that computer science actually concerns the study of data structures and algorithms that can be implemented in any programming language, a study which is more profound and elusive than striving for fluency in as many programming languages as possible.

This book has covered many topics in the field of Natural Language Processing. Most of the examples have used Python and English. However, it would be unfortunate if readers concluded that NLP is about how to write Python programs to manipulate English text, or more broadly, about how to write programs (in any programming language) to manipulate text (in any natural language). Our selection of Python and English was expedient, nothing more. Even our focus on programming itself was only a means to an end: as a way to understand data structures and algorithms for representing and manipulating collections of linguistically annotated text, as a way to build new language technologies to better serve the needs of the information society, and ultimately as a pathway into deeper understanding of the vast riches of human language.

But for the present: happy hacking!