

Book Review

Speech and Language Processing (second edition)

Daniel Jurafsky and James H. Martin

(Stanford University and University of Colorado at Boulder)

Pearson Prentice Hall, 2009, xxxi+988 pp; hardbound, ISBN 978-0-13-187321-6, \$115.00

Reviewed by

Vlado Keselj

Dalhousie University

Speech and Language Processing is a general textbook on natural language processing, with an excellent coverage of the area and an unusually broad scope of topics. It includes statistical and symbolic approaches to NLP, as well as the main methods of speech processing. I would rank it as the most appropriate introductory and reference textbook for purposes such as an introductory fourth-year undergraduate or graduate course, a general introduction for an interested reader, or an NLP reference for a researcher or other professional working in an area related to NLP.

The book's contents are organized in an order corresponding to different levels of natural language processing. After the introductory chapter 1, there are five parts:

- Part I, Words: five chapters covering regular expressions, automata, words, transducers, n -grams, part-of-speech tagging, hidden Markov models, and maximum entropy models.
- Part II, Speech: five chapters covering phonetics, speech synthesis, recognition, and phonology.
- Part III, Syntax: five chapters covering a formal grammar of English, syntactic and statistical parsing, feature structures and unification, and complexity of language classes.
- Part IV, Semantics and Pragmatics: five chapters covering representation of meaning, computational semantics, lexical and computational lexical semantics, and computational discourse.
- Part V, Applications: four chapters covering information extraction, question answering, summarization, dialogue and conversational agents, and machine translation.

The first edition of the book appeared in 2000 with the same title, and a very similar size and structure. The structure has been changed by breaking the old part "Words" into two parts "Words" and "Speech," merging two old parts "Semantics" and "Pragmatics" into one "Semantics and Pragmatics," and introducing one new part "Applications." I considered the old edition also to be the textbook of choice for a course in NLP, but even though the changes may not appear to be significant, the new edition is a marked improvement, both in overall content structure as well as in presenting topics at a finer-grained level. Topics on speech synthesis and recognition are significantly expanded; maximum entropy models are introduced and very well

explained; and statistical parsing is covered better with an explanation of the principal ideas in probabilistic lexicalized context-free grammars.

Both editions include very detailed examples, with actual numerical values and computation, explaining various methods such as n -grams and smoothing. As another example, maximum entropy modeling is a popular topic but in many books explained only superficially, while here it is presented in a well-motivated and very intuitive way. The learning method is not covered, and more details about it would be very useful. The new edition conveniently includes the following useful reference tables on endpapers: regular expression syntax, Penn Treebank POS tags, some WordNet 3.0 relations, and major ARPAbet symbols.

The book was written with a broad coverage in mind (language and speech processing; symbolic and stochastic approaches; and algorithmic, probabilistic, and signal-processing methodology) and a wide audience: computer scientists, linguists, and engineers. This has a positive side, because there is an educational need, especially in computer science, to present NLP in a broad, integrated way; this has seemed to be always very challenging and books with wide coverage were rare or non-existent. For example, Allen's (1995) *Natural Language Understanding* presented mostly a symbolic approach to NLP, whereas Manning and Schütze's (1999) *Foundations of Statistical Natural Language Processing* presented an exclusively statistical approach. However, there is also a negative side to the wide coverage—it is probably impossible to present material in an order that would satisfy audiences from different backgrounds, in particular, linguists vs. computer scientists and engineers.

In my particular case, I started teaching a graduate course in Natural Language Processing in 2002 at Dalhousie University, which later became a combined graduate/undergraduate course. My goal was to present an integrated view of NLP with an emphasis on two main paradigms: knowledge-based or symbolic, and probabilistic. Not being aware of Jurafsky and Martin's book at the time, I was using Manning and Schütze's book for the probabilistic part, and Sag and Wasow's (1999) book *Syntactic Theory: A Formal Introduction* for the symbolic part. I was very happy to learn about Jurafsky and Martin's book, since it fitted my course objectives very well. Although I keep using the book, including this new edition in Fall 2008, and find it a very good match with the course, there is quite a difference between the textbook and the course in order of topics and the overall philosophy, so the book is used as a main supportive reading reference and the course notes are used to navigate students through the material. I will discuss some of the particular differences and similarities between Jurafsky and Martin's book and my course syllabus, as I believe my course is representative of the NLP courses taught by many readers of this journal.

The book introduces regular expressions and automata in Chapter 2, and later introduces context-free grammars in Chapter 12, followed by some general discussion about formal languages and complexity in Chapter 16. This is a somewhat disrupted sequence of topics from formal language theory, which should be covered earlier in a typical undergraduate computer science program. Of course, it is not only a very good idea but necessary to cover these topics in case a reader is not familiar with them; however, they should be presented as one introductory unit. Additionally, a presentation with an emphasis on theoretical background, rather than practical issues of using regular expressions, would be more valuable. For example, the elegance of the definition of regular sets, using elementary sets and closure of three operations, is much more appealing and conceptually important than shorthand tricks of using practical regular expressions, which are given more space and visibility. As another example, it is hard to understand the choice of discussing equivalence of deterministic

and non-deterministic finite automata in a small, note-like subsection (2.2.7), yet giving three-quarters of a page to an exponential algorithm for NFSA recognition (in Figure 2.19), with a page-long discussion. It may be damaging to students even to mention such a poor algorithm choice as the use of backtracking or a classical search algorithm for NFSA acceptance. Context-free grammars are described in subsection 12.2.1; besides the need to have them earlier in the course, actually as a part of introductory background review, more space should be given to this important formalism. In addition to the concepts of derivation and “syntactic parsing,” the following concepts should be introduced as well: parse trees, left-most and right-most derivation, sentential forms, the language induced by a grammar, context-free languages, grammar ambiguity, ambiguous sentences, bracketed representation of the parse trees, and a grammar induced by a treebank. Some of these concepts are introduced in other parts of the book. More advanced concepts would be desirable as well, such as pumping lemmas, provable non-context-freeness of some languages, and push-down automata.

As noted earlier, the order of the book contents follows the levels of NLP, starting with words and speech, then syntax, and ending with semantics and pragmatics, followed by applications. From my perspective, having applications at the end worked well; however, while levels of NLP are an elegant and important view of the NLP domain, it seems more important that students master the main methodological approaches to solving problems rather than the NLP levels of those problems. Hence, my course is organized around topics such as n -gram models, probabilistic models, naive Bayes, Bayesian networks, HMMs, unification-based grammars, and similar, rather than following NLP levels and corresponding problems, such as POS tagging, word-sense disambiguation, language modeling, and parsing. For example, HMMs are introduced in Chapter 5, as a part of part-of-speech tagging; language modeling is discussed in Chapter 4; and naive Bayes models are discussed in Chapter 20.

The discussion of unification in the book could be extended. It starts with feature structures in Chapter 15, including discussion of unification, implementation, modeling some natural language phenomena, and types and inheritance. The unification algorithm (Figure 15.8, page 511) is poorly chosen. A better choice would be a standard, elegant, and efficient algorithm, such as Huet’s (e.g., Knight 1989). The recursive algorithm used in the book is not as efficient, elegant, nor easy to understand as Huet’s, and it contains serious implementational traps. For example, it is not emphasized that the proper way to maintain the pointers is to use the UNION-FIND data structure (e.g., Cormen et al. 2002). If the pointers f_1 and f_2 are identical, there is no need to set $f_1.pointer$ to f_2 . Finally, if f_1 and f_2 are complex structures, it is not a good idea to make a recursive call before their unification is finished, since these structures may be accessed and unified with other structures during the recursive call. The proper way to do it is to use a stack or queue (usually called sigma) in Huet’s style, add pointers to structures to be unified on the stack, and unify them after the unification of current feature structure nodes is finished. Actually, this is similar to the use of “agenda” earlier in the book, so it would fit well with previous algorithms.

Regarding the order of the unification topics, I prefer an approach with a historical order, starting from classical unification and resolution, followed by definite-clause grammars, and then following with feature structures. The Prolog programming language is a very important part in the story of unification, and should not be skipped, as it is here. More could be written about type hierarchies and their implementation, especially because they are conceptually very relevant to the recent popular use of ontologies and the Semantic Web.

As a final remark on the order, I found it useful in a computer science course to present all needed linguistic background at the beginning, such as English word classes (in Chapter 5), morphology (in Chapter 3), typical rules in English syntax (in Chapter 12), and elements of semantics (in Chapter 19), and even a bit of pragmatics. As can be seen, these pieces are placed throughout the book. The introduction of English syntax in Chapter 12 is excellent and better than what can be typically found in NLP books, but nonetheless, the ordering of the topics could be better: Agreement and other natural language phenomena are intermixed with context-free rules, whereas in my course those two were separated. The point should be that context-free grammars are a very elegant formalism, but phenomena such as agreement, movement, and sub-categorization are the issues that need to be addressed in natural languages and are not handled by a context-free grammar (cf. Sag and Wasow 1999).

I also used the textbook in a graduate reading course on speech processing, with emphasis on speech synthesis. The book was a useful reference, but the coverage was sufficient for only a small part of the course.

The following are some minor remarks: The title of Chapter 13, "Syntactic Parsing," is unusual because normally parsing is considered to be a synonym for syntactic processing. The chapter describes the classical parsing algorithms for formal languages, such as CKY and Earley's, and the next chapter describes statistical parsing. Maybe a title such as "Classical Parsing," "Symbolic Parsing," or simply "Parsing" would be better. The Good-Turing discounting on page 101 and the formula (4.24) are not well explained. The formula (14.36) on page 479 for harmonic mean is not correct; the small fractions in the denominator need to be added.

In conclusion, there are places that could be improved, and in particular, I did not find that the order of material was the best possible. Nonetheless, the book is recommended as first on the list for a textbook in a course in natural language processing.

References

- Allen, James. 1995. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA.
- Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., and Stein, Clifford. 2002. *Introduction to Algorithms*, 2nd edition. The MIT Press, Cambridge, MA.
- Knight, Kevin. 1989. Unification: A multidisciplinary survey. *ACM Computing Surveys*, 21(1): 93–124.
- Manning, Christopher D. and Schütze, Hinrich. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA.
- Sag, Ivan A. and Wasow, Thomas. 1999. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, CA.

Vlado Keselj is an Associate Professor of the Faculty of Computer Science at Dalhousie University. His research interests include natural language processing, text processing, text mining, data mining, and artificial intelligence. Keselj's address is: Faculty of Computer Science, Dalhousie University, 6050 University Ave, Halifax, NS, B3H 1W5 Canada; e-mail: vlado@cs.dal.ca.