Book Reviews

Computer Processing of Natural Language

Gilbert K. Krulee
(Northwestern University)

Englewood Cliffs, NJ: Prentice Hall,
1991, vii + 456 pp.
Hardbound,
ISBN 0-13-610288-3, price not listed

Computers and Human Language

George W. Smith
(University of Massachusetts, Boston)

New York: Oxford University Press,
1991, xiv + 478 pp.
Hardbound,
ISBN 0-19-506281-7, price not listed;
paperbound, ISBN 0-19-506282-5, $16.95

Reviewed by
Laura Proctor
City Polytechnic of Hong Kong

Krulee's *Computer Processing of Natural Language* is a text based on a course offered by the Department of Electrical Engineering and Computer Science at Northwestern University. Krulee, holding joint appointments in Computer Science and Linguistics, summarizes his perspective in this quotation from the introductory chapter (p. 1):

> This is a book about linguistic analysis involving a pair of issues that complement each other. First, there is an emphasis on representations: on grammars for describing and generating the sentences that make up a given language. Second, there is an emphasis on processing and computation: on demonstrating that the grammar for a language has important implications for how it can be processed. More specifically, we will be discussing algorithms that describe systematic relationships between the grammar for a given language and the ability to process sentences that belong to it.

The material in the text is structured in keeping with these two issues, beginning with descriptions of languages (Chapters 1–3) and then proceeding to methods of computational implementation of those descriptions (Chapters 4–6).

The major headings in the table of contents (and number of pages in each) are as follows:

1. Introduction (44 pages)
2. Transformational grammars for natural language (57 pages)
3. Two-level representations (65 pages)
4. Transition networks: From grammar to acceptor (52 pages)
5. Two-level processing systems (80 pages)
6. Meaning and interpretation (73 pages)
7. Issues and applications (48 pages)

Each chapter begins with a paragraph that presents general issues or questions relevant to the material in the chapter, followed by a brief overview of the chapter relating its content to that of preceding sections. At the end of the chapters, there is a summary of the material presented, and a further discussion relating it in more detail to the issues and questions introduced at the beginning. A final section provides suggestions for further readings that address issues presented in the text in greater detail and guide the reader to alternative approaches not discussed in the text itself. The presentation in each chapter includes very detailed examples worked out to demonstrate the application of the formally described grammar or processing method. Because of the detail in these examples, the book must be read slowly and carefully, since the examples are an integral part of the exposition. Although the methods of description and processing are presented in detail, there is no discussion of methods of coding in a particular programming language.

Chapter 1, following an overview of the contents of the text, reviews basic notions from formal language theory (i.e., formal grammars, the Chomsky hierarchy, derivation trees, equivalence of grammars, recursion, and decidability). For "computer scientists, particularly those with some exposure to the theory of programming languages, the first portion of this book will seem like a review of familiar materials... very much in the style of Aho and Ullman's *Theory of parsing, translation and compiling* (1972)." The second chapter describes the transformational approach to language description (following the "standard theory" of Chomsky 1965), with specific examples drawn from English syntactic structure. In his chapter summary, Krulee reviews the strengths and weaknesses of the theory in terms of both description and computation, which leads into the third chapter where a two-level grammar formalism (drawing upon the formalism used to define Algol 68) is introduced. The main feature of this formalism is the use of meta-variables and meta-grammar as an alternative to the explicit use of transformations.

Following his discussion of sentence grammars as descriptions, the emphasis in Chapters 4, 5, and 6 turns to issues of processing. In Chapter 4, the construction of acceptors for right-linear and context-free grammars is demonstrated, introducing both simple and recursive transition networks. The concepts of determinism, backtracking, and the relation between derivation sequence (associated with the grammar) and accepting sequence (associated with the acceptor) are all introduced and illustrated through practical examples. Chapter 5 then introduces, in detail, augmented transition networks as acceptors/parsers for the two-level grammars presented in Chapter 3, investigating properties of particular grammars that lead to nondeterministic processing and the need for backtracking or look-ahead. A final section discusses Earley's parsing method as another approach to dealing with ambiguous grammars.

In Chapter 6, Krulee shifts the focus from processing syntax to consider the need for semantic and pragmatic interpretation in real applications of natural language processing. The example he chooses is a natural language interface to a database management system. Logic as a form of semantic representation for statements and questions in natural language is briefly introduced. A syntax-directed translation strategy, based on the output of parsing methods introduced in the preceding chapters, that maps syntactic structures onto a semantic representation, is introduced and illustrated through several detailed examples. A short discussion of Prolog and rules of inference is included to suggest how pragmatic analysis might be incorporated into an application system. The final chapter strikes a more general note, briefly discussing particular applications (machine translation, question answering, and expert systems) and some of the issues they have raised in the area of natural language processing.

At the end of the text, there are exercises (about 10 for each chapter) included that suggest extensions to examples in the text or additional data to which grammars and processors could be applied. The exercises for Chapter 7 suggest readings that could be evaluated in light of issues raised in the text rather than practical exercises, in keeping with the tone of the chapter itself. There are no suggested solutions to any of these exercises.

The formal presentation and detailed examples, particularly in the first five chapters, serve to give an in-depth look at one approach to processing natural language on a computer. This depth of analysis is the book's strength. The language structures addressed go well beyond simple sentential forms, providing a realistic picture of the complexity of any natural language and the relation between grammatical descriptions and processing methods is illustrated in detail. The text does not provide a general view of current linguistic formalisms. Transformational grammar, as acknowledged by the author, was not chosen for its currency but rather as a "convenient" starting point for discussing descriptions of natural language. (In the suggestions for further readings at the end of Chapter 3, a number of more recent developments in linguistic theory are mentioned, and references to relevant literature is provided.) Although the two-level formalism presented as an alternative to transformations usefully illustrates the advantages of meta-variables and features for language processing, the connection to more current developments in linguistic theory is not explicitly discussed.

George Smith begins his preface to *Computers and Human Language* with the following words:

> What do you know when you know a language? Why is it that words form sentences in certain combinations but not others? How do words and sentences mean? What takes place in the instant needed to comprehend an utterance? This book is about computational approaches to such questions.

As the quotation suggests, this textbook is not designed to teach students how to write computer programs to process human language, nor is it a description of computational research projects. Rather, it focuses on the ways in which the use of computers and the application of computational methods have influenced the study of language over the past forty years. It provides an overview of the ways computers have been used to analyze linguistic information and use linguistic knowledge to emulate or simulate language understanding. The author not only directs the reader's attention to computational representations and processes, but also makes sure that the linguistic and psychological motivations are clear. He presents issues, problems, and potential solutions along with their successes and shortcomings. Throughout the text, formal linguistic analyses, computational methods, and psychological research are blended together to provide a broad picture of the many facets of language and the challenges of processing language with computers. All concepts are carefully explained with intuitive examples that make the basic principles very accessible to students with little or no previous background in the area of computational approaches to linguistics. The text is notable for its breadth of coverage, including consideration of both written and spoken language, ranging from encoding orthographies and sounds through word formation rules, lexical organization, syntactic and semantic processing to discourse interpretation and knowledge representation. Given this breadth, none of the topics is presented in detail; however, the treatment is not trivial. Plenty of actual language examples are included to illustrate linguistic phenomena, and simple examples of processing algorithms illustrate the basic principles of their applications.

The material is organized according to the traditional levels of linguistic inquiry; beginning with letters and phonemes and proceeding to discourse. Overlaid on this structure the first chapters introduce concepts of representation, data structures, and processing algorithms for students with no background in computing to lay the groundwork for understanding the later chapters. The major headings of the table of contents are listed below, along with the length of each chapter, to provide a general picture of the text's content.

1. Components of words (30 pages)
2. The challenge of spoken language (28 pages)
3. Words and the lexicon (30 pages)
4. Structure and search (29 pages)
5. Sublexical and lexical processing in parallel (25 pages)
6. Approaches to syntax (50 pages)
7. Augmented parsers and modern grammars (53 pages)
8. Lexical semantics (48 pages)
9. Phrase and sentence semantics (35 pages)
10. Integrating syntactic and semantic processing (37 pages)
11. Discourse interpretation using world knowledge (29 pages)
12. Knowledge about discourse (30 pages)

The first chapter introduces the notion of symbolic systems and coding schemes used to represent written texts according to their orthography. Several elementary algorithms based on English morphological rules are introduced to illustrate methods for processing strings. Chapter 2 then turns to spoken language, beginning with a simple discussion of phonemes and phonotactics before addressing methods of acoustic analysis and approaches to speech recognition and synthesis. The importance of context provided by other levels of linguistic organization (syntax, semantics, discourse) in speech processing is included in the discussion.

Moving to the next level, words, Chapter 3 gives a brief description of lexical content and structure (anticipating a more thorough discussion in Chapter 8). The application of computers to analysis of word lists and concordancing is used to introduce procedures for sorting, indexing, and statistical analysis of corpora. In these first three chapters, the discussion of data structures and processing methods is informal, the emphasis placed on establishing their relevance in the context of linguistic analysis.

Chapters 4 and 5 take a more formal approach to the introduction of data structures and processing. First, the more standard material including linear and linked lists, trees and network representations, and techniques for search and retrieval are addressed. Then, in Chapter 5, connectionist systems are presented. In both chapters, specific applications in lexical and perceptual processing are used to illustrate the structures presented.

Beginning with Chapter 6, the focus shifts to developing a model for human language processing, giving careful attention to linguistic data and psychological research in order to clarify the motivation for the computational methods that are explored. In Chapters 6 and 7, the topic is syntactic processing. Beginning from a simple model of sentences as linked lists of words, Smith proceeds to introduce the notion of grammatical categories and context-free phrase structure grammars. From this point, he discusses the relation between grammar and parsing methods, covering topics such as recursion, bottom-up versus top-down strategies, backtracking, and chart parsing, all related to specific linguistic phenomena and observed human processing. Chapter 7 continues the discussion by introducing agreement phenomena, such as number and pronomial case in English, as motivation for ATNs and unification strategies. The chapter ends with a brief discussion of GPSG (its metarules and use of features) and LFG as alternative formalisms.

Semantic representation and processing are the topic of the next two chapters. Chapter 8 covers issues related to lexical representation: relating propositions and predicate calculus to network representations; surveying approaches to lexical representation such as meaning postulates, semantic features, and selectional restrictions; and contrasting prototypical and referential meaning. Chapter 9 goes on to the representation and processing of phrase and sentence meanings using a relational representation, relating grammatical functions to semantic cases, and introducing conceptual dependency. Chapter 10, as its title indicates, discusses the integration of syntactic and semantic processing. After presenting motivations for integration and a discussion of the issue of modularity, the author provides some fairly detailed examples to highlight both the advantages and problems with methods for interpretation of sentences based on semantics only versus those that interleave syntax and semantics.

The last two chapters introduce some of the problems and approaches to interpreting discourse; that is, units of language greater than one sentence, whether written or spoken. Some of the topics touched upon are: general knowledge and inferences, scripts, plans, discourse segmentation, and resolution of anaphora.

There are some suggested exercises at the end of every chapter (though no suggested answers). These suggest additional data or variations to processing schemes to encourage students to explore the possibilities and limitations of the methods introduced in the chapter. Each chapter includes a "Further reading" section that directs the reader to items included in the bibliography (27 pages) that are relevant to the major topics covered in the chapter. The bibliography itself is current and provides references to works in linguistics, psychology, and computing related to language processing. There is also an index that includes major topics, terms, and authors used or referenced in the text.

This text can be described as a guided tour through the forest of computational approaches to the study of language, pointing out many of the fascinating trees that deserve further examination. Drawing on developments in the fields of linguistics, psychology, and computing, it succeeds in giving a good overview of problems and issues in language processing and provides a foundation for understanding how the fields interact with and contribute to each other. Its strength is in focusing on the connections between levels and methods of processing, something that is often difficult to see through reading the literature of each individual field.

The two texts reviewed here address very different audiences. Krulee's text focuses on issues related to computation (primarily syntactic processing) rather than general linguistic applications. Since the text does not provide an insight into modern linguistic theory for students in computer science, and it leaves linguistics students largely on their own to find the relation between the proposed formalism and those that they may have studied, it is not appropriate as an introduction to computational linguistics for students in either field. But it does provide a detailed treatment of grammatical formalism and related processing methods in relation to natural languages, making it suitable for advanced undergraduate or graduate students in computing science (or linguistics students with a very serious interest in natural language processing applications). Smith's text, on the other hand, fills the gap felt by this reviewer in providing a basic, nontechnical text for an introductory survey course for undergraduate students in linguistics that covers the full range of linguistic issues from a computational perspective, including representation of orthography and speech processing, which are either assumed or ignored by other texts. Because of its greater breadth and more extensive examples, I would prefer this text to Grishman's (1986) introduction. For computer science students, the book can provide a comprehensive overview of the complexity of human language from a linguistic perspective.

References

Aho, Alfred V., and Ullman, Jeffrey D. (1972). *Theory of Parsing, Translation, and Compiling. Volume I: Parsing*. Englewood Cliffs, NJ: Prentice Hall.

Chomsky, Noam (1965). *Aspects of the Theory of Syntax*. Cambridge, MA: The MIT Press.

Grishman, Ralph (1986). *Computational Linguistics: An Introduction.* Cambridge, England: Cambridge University Press.

Laura Proctor received her B.Sc. from the University of Guelph in 1975 and worked for ten years in technical support and user training primarily at the University of Victoria (Canada) before returning to academic study. In 1990, she received her M.A. in Linguistics from the University of Victoria, where she participated in the development and teaching of computing courses offered by the Department of Linguistics. She is now teaching computational linguistics in the Department of Applied Linguistics at the City Polytechnic of Hong Kong.