Text Generation and Systemic-Functional Linguistics: Experiences from English and Japanese

Christian M. I. M. Matthiessen and John A. Bateman
(University of Sydney and Gesellschaft für Mathematik und Datenverarbeitung, Darmstadt)

London: Pinter Publishers
(Communication in Artificial
Intelligence Series), 1991, xxii + 348 pp.
Hardbound, ISBN 0-86187-711-X, £52.50

Reviewed by
Terry Patten
The Ohio State University

This is an important book. On the surface, it is a survey and summary of work related to a major computational linguistics effort—the Penman project at the University of Southern California. Beneath the specific title and the apparently narrow subject matter, however, lies a general point about the field of computational linguistics: many important aspects of language are not addressed by the generative tradition that has dominated the field. This book aims to demonstrate that other types of linguistic description are available to serve as the basis for a computational linguistic treatment of these issues.

As the second half of the book's title suggests, the research presented is based on M. A. K. Halliday's functional theory of grammar (see Winograd [1983, Chapter 6] for an excellent computationally oriented synopsis). Some confusion has surrounded Halliday's work and the notion of a functional grammar, as indicated by another recent review (Fraser 1991, pp. 104–106):

> Halliday's Systemic (Functional) Grammar ... is an example of a functional theory that has served as the basis for some interesting computer systems, especially those concerned with text generation. This is probably less attributable to the functional credentials of Systemic Grammar than to its exceptionally useful descriptive tool, the system network .... [W]hat, if any, are the distinctive benefits of functional theories ... for NLP?

In fact, the usefulness of the system networks lies primarily in their ability, as a classification formalism, to provide functional rather than structural descriptions. It is no coincidence that Systemic Grammar and other functional descriptions of language (e.g., Functional Unification Grammar; see Appelt [1983]) are widely used in text generation (the first half of the title currently under review). The question of their distinctive benefits is, to a large extent, what Matthiessen and Bateman's book is all about.

The book's argument for functional description is made primarily in the context of the Penman project. Penman has involved designing a general architecture for text generation. Certainly a large part of such an effort must be providing a realization component that contains adequate grammatical and lexical coverage. But a large grammar and lexicon will allow many different utterances to convey the same intended meaning, and a general architecture must provide a mechanism to decide which is best. A link is required between the grammatical and lexical knowledge on one hand and

socio-pragmatic and discourse knowledge that can make these decisions on the other. By indexing the grammatical knowledge by function, Systemic (and other functional) Grammars facilitate making this link.

The book is organized into four parts. The first part provides an overview of the process of text generation and sketches the general architecture of the Penman system. Some of the issues faced by generation systems are summarized, and some other generation systems are briefly described. The second part of the book goes into the details of Systemic Grammar and computational implementations of Systemic Grammar for both English and Japanese generation. Several interesting computational linguistic issues are discussed, including discourse constraints in Japanese that illustrate the need to worry about more than conveying propositional content when generating text. The third part is really the core of the book. It describes the issues that arose when implementing Halliday's theory. To a large extent, these issues center upon the interactions between the grammar and the higher-level knowledge upon which grammatical decisions depend. The examples in this part of the book involving politeness in Japanese are particularly interesting and call for increased computational attention to socio-pragmatic issues. The fourth and final part of the book is a discussion of unresolved general issues. This part begins with a short chapter on parallel processing that might well have been omitted from the book. The next chapter of this part concerns procedural and declarative interpretation of system networks, and is reviewed further below. The subsequent chapter offers a good discussion of the computational role of contextual knowledge in generation.

The layout of the book is done well. The only significant typographical problem is the lack of a systematic treatment of feature names; these occasionally become confusing when appearing mid-sentence with no special indication.

It is important to note that the book is written from the start using Hallidayan terminology, and this overlaps to a surprisingly small degree with the terminology typical of computational linguistics. As a result, the readability of the book is reduced significantly for much of the potential audience. The authors very wisely have appended a glossary covering a good portion of this terminology (as well as some of the computational terminology for the benefit of non-computational linguists). Nevertheless, readers not familiar with systemic theory would be well advised to obtain and carefully review Winograd's excellent synopsis of systemic grammar (op. cit.) before beginning Matthiessen and Bateman's book.

The only technical criticism of the book worth mentioning here concerns the interpretation of the "system networks." These networks are formally classification hierarchies, but historically there has been a tendency among systemic linguists to informally interpret them as some sort of flowchart, and talk about making "choices" while traversing the networks. As Winograd (1983, pp. 278–279) cautions:

> In describing the role of classification in systemic grammar, we have carefully been using words like 'classify,' 'alternatives,' and 'describe,' avoiding the term 'choice,' which is more commonly used. This was to emphasize the fact that the classification is being imposed by an observer—by someone who is describing the form and use of utterances. One can talk about the classification of animals by a biologist, but it hardly seems appropriate to talk about a system of 'choices' in the sense of an active choosing.

Failing to distinguish between a declarative hierarchy that describes a classification scheme and a process that performs a classification task has led to much confusion

in the systemic literature. Unfortunately, this book, in general, makes a hash of procedural and declarative description. The chapter on systemic theory states (p. 81) that the theory must describe procedures for processing the networks—but this is no more appropriate than stating that a biological classification must include such procedures. Chapters describing early work by Matthiessen and Bateman use an explicitly procedural interpretation of the networks, even allowing loops for iteration. Chapter 11, in contrast, explicitly attempts to provide an alternative to these loops and clarify the procedural/declarative distinction. Unfortunately, it provides a vague discussion of the alternative, and still seems to blur the distinction to some extent. In any case, the main contribution of the book—which stems from the content of the system networks and the interaction of that content with higher-level knowledge—is not significantly diminished.

In the final analysis, this book is a description of interesting and important text generation work. For the reader less interested in generation, the book offers a good discussion of language issues not widely addressed in the computational literature (e.g., register, politeness) and insight into a linguistic theory—Systemic Functional Grammar—that shows some promise as a basis for computational treatment of these types of issues.

This is a book that will demand considerable effort from its readers. It addresses issues that will be familiar to few, in terms that will be familiar to fewer. But while looking at new problems from a new perspective makes this book difficult, it also makes this book valuable.

References

Appelt, Douglas E. (1983). "TELEGRAM: A grammar formalism for language planning." *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, 595–599.

Fraser, Norman (1991). Review of: Connolly, J. H. and Dik, S., (editors). *Functional Grammar and the Computer*, 1989. *Computational Linguistics*, **17**(1), 104–106.

Winograd, Terry (1983). *Language as a Cognitive Process, Volume I: Syntax*. Addison-Wesley.

Terry Patten is an assistant professor of Computer and Information Science at The Ohio State University. He received his Ph.D. from the Department of Artificial Intelligence, University of Edinburgh in 1986. His address is CIS, 2036 Neil Avenue Mall, Columbus, OH 43210; e-mail: patten@cis.ohio-state.edu