

- Gazdar, Gerald and Pullum, Geoffrey 1985 Computationally Relevant Properties of Natural Languages and Their Grammars. *New Generation Computing* 3; 273–306.
- Higginbotham, James 1985 On Semantics. *Linguistic Inquiry* 16; 547–593.
- Hirst, Graeme 1987 *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge, England.
- Pullum, Geoffrey 1983 Context Freeness and the Computer Processing of Human Languages. In *Proceedings of the 21st Meeting of the Association for Computational Linguistics*, Cambridge, MA: 1–6.
- Pullum, Geoffrey 1986 A Guest of the State. *Natural Language and Linguistic Theory* 4; 284–289.
- Putnam, Hilary 1975 The Meaning of 'Meaning'. In *Mind, Language, and Reality: Philosophical Papers 2*, Cambridge University Press, Cambridge, England.

NOTES

1. The title shown above is that given on the title page and back cover, but its listing with the Library of Congress is the slightly more appropriate *Philosophy, Language, and Artificial Intelligence: Philosophical Resources for Natural Language Processing*.
2. Not only was the need to pay royalties avoided, but typesetting expenses seem to have been spared as well. For the most part the essays appear as they were originally typeset, pagination being the only change. This policy is unfortunate, for mistakes have surfaced in some of these writings which ought to be noted in editorial footnotes, if not corrected. For example, there is a slip in Montague's paper which is reprinted without mention here (the first meaning postulate on p. 157 contains a biconditional instead of a conditional—fine in the case of intransitive verbs, but not for common nouns). This "bug" has infested some implementations of Montague grammar.
3. Perhaps, under the circumstances, I can be forgiven for citing a "Topic . . . comment" column myself.

This oasis, this last respite from the seriousness of professionalized linguistics, this space reserved at the end of each *NLLT* for a piece of writing flippant and inconsequential enough to give an exhausted assistant professor the strength to get up and do what needs to be done, had been mistaken for mere research, and was being indexed and abstracted (Pullum 1986, p. 288).

Peter Ludlow is an assistant professor of philosophy. His current research interests are in the philosophy of language and of cognitive science. He has previously worked on projects in natural language processing for Honeywell's Corporate Systems Development Division. Ludlow's address is: Department of Philosophy, SUNY at Stony Brook, Stony Brook, NY 11794. E-mail: PLUDLOW@SBCCVM.bitnet

SYSTEMIC TEXT GENERATION AS PROBLEM SOLVING

Terry Patten

(Department of Computer and Information Science,
The Ohio State University)

Cambridge University Press, 1988, xii + 214 pp.
(Studies in Natural Language Processing)
ISBN 0-521-35076-X, \$34.50 (20% discount to ACL
members) (hb)

Reviewed by
Eduard Hovy

USC Information Sciences Institute

In thinking about the problem of generating English sentences, it is very reasonable to wonder whether the natural paradigm of the solution is not perhaps that of planning or problem solving. (I use the terms here interchangeably; the distinctions between the two enterprises are small enough to be ignored in this context.) That is to say, given that you start with some specific state, namely the input, and want to end with some other state, the corresponding sentence, and you have a set of rules, namely the grammar, that prescribe how to bring about the change, then why not simply use a planner or problem solver to do the job?

This is exactly what Terry Patten's book is about. It does not address all the issues involved, but it shows convincingly that you can implement a sentence generator in the problem-solving paradigm and it provides as well some very useful information on systemic grammar: both a formal definition of systemic networks and a description of a particular implementation of one.

The book consists of three parts: introductory material, the core idea, and the rest. The introductory material contains a chapter describing AI problem solving and a chapter describing systemic grammar. The core material first describes how one can interpret a grammar as information with which one can perform problem solving, and then provides a formal model of systemic grammar. In the remainder, a particular implementation of a problem solver using systemic grammar, called SLANG, is described and compared with other generators, and parts of SLANG's grammar and samples of its output are provided.

Chapter 3, the background chapter describing systemic grammar, is a simple introduction to a body of thought that often has been called impenetrable. Given the breadth of application of systemic linguistics, an understanding of it is required for anyone who wishes to venture beyond the narrow view of language as taken by the various generative paradigms. Since it focuses on implementational issues, this chapter offers, specially tailored for the computational linguistics community, one of the most readable descriptions of the systemic view of language I have yet encountered. Unfortunately, this fact makes the chapter less suitable as a general overview of the field, since it gives no indication of the depth of linguistic research that underlies the ideas it describes.

Chapter 4 leads the reader through the correspondences between problem solving, which is a search process through a space of alternative states, and systemic language generation, which is a search process through a network of alternative meanings (i.e., aspects of sentences). The network's grammatical choice points, called systems, are implemented by production rules. In the way illustrated, AI techniques for handling huge spaces of interdependent alternatives efficiently can be applied to fine-grained grammatical distinctions identified by systemic (or other functional) linguists.

Chapter 5 contains a somewhat exploratory formal

model of systemic grammar (to my knowledge the only such formal model extant today). This model includes a formal definition of the syntactic structures and hence the sentences generated by the grammar. Realization rules (the rules by which the set of features, computed by the network traversal, are assembled into an English sentence) are not fully treated, though an outline is given of how they could be formulated in a first-order sorted logic.

Chapter 6 describes the implementation of SLANG, and Appendix B contains some sample sentences it generated. Appendix A contains a tutorial of the OPS5 production language.

One serious shortcoming mars the overall message of the book. This is the inadequate treatment of presentational planning issues. It has become increasingly clear over the last decade that realization—i.e., the formulation of smallish chunks of information as individual sentences—is but one half of the problem of generating language, the small half at that. The other half involves the collection and organization of material into such chunks, a much less well-understood problem.

This is where the book disappoints. Patten's approach is to use the systemic notion of register—that is, roughly, the characteristics of a particular interaction between interlocutors, situated in time and place—to control the problem-solver-generator by setting crucial choice point values in the grammar network. This operation is called preselection. Once these values have been set, the remaining features are all fully determined and the production rules take care of their collection. Leaving aside the fact that Patten's interpretation of register is highly controversial in systemic linguistics, its use in the book significantly weakens the force of his claims. Naturally, one expects an account of the process of constructing various registers and of determining which points in the grammar registers should minimally control (or if not, how one is to go about making the remaining problem-solving decisions); one expects to find if not a full theory of register, at least a description of the theory underlying register. Almost no such information is given; instead, appeal is made to some text planner to construct, using information on the "semantic stratum", the appropriate values for preselection of appropriate values. Fragments of examples of semantic strata are given, one for a mother-child interaction and one, in an appendix, for a building task, but many questions about their status remain unaddressed: What is a semantic stratum? What information can it contain? How complete are the given strata? Do all mother-child interactions obey the given distinctions? Of the thousands of other locutions with which the mother could chastise and threaten, why are precisely the given sentences generated? What would control further variation? And so forth. The lack of development of a notation with which to control generation—an input notation, in some sense—is a serious shortcoming.

The book's strong points are, therefore, its accessible introduction to systemic grammar, its demonstration that a grammar can be implemented by production rules and treated in a problem-solving paradigm, its formal model of a systemic grammar, and the OPS5 tutorial. The reader will come from the book having learned something about all these matters, though not necessarily convinced that it describes the best paradigm for generation. Still, it is a valuable addition to any collection of books about the art of generating sentences by computer.

Eduard Hovy is a research scientist at the Information Sciences Institute of the University of Southern California. He is a member of the Penman natural language generation project, working on text planning. He completed a Ph.D. at Yale University in 1987 on the generation of language under the guidance of pragmatic goals. His address is: Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292-6695. E-mail: hovy@venera.isi.edu

BRIEFLY NOTED

INTEGRATED NATURAL LANGUAGE DIALOGUE: A COMPUTATIONAL MODEL

Robert E. Frederking
(Siemens, A.G.)

Boston: Kluwer Academic Publishers, 1988, xvii + 178 pp.
(The Kluwer International Series in Engineering and
Computer Science; Natural Language Processing and
Machine Translation)
ISBN 0-89838-255-6, \$39.95 (hb)

Frederking describes a prototype natural language interface with two special features. First, the system is integrated in the sense that all processing—syntactic, semantic, pragmatic—is performed in the one module, written as a set of production rules. Second, there is particular attention to the resolution of elliptical utterances in the dialogue.

Frederking claims that "no other system can handle the range of ellipsis phenomena" (p. 5) that his can, though this is not backed up with any point-by-point comparison with other research. For example, the extensive work of the Linguistic String Project (e.g., Sager 1981; Sager, Friedman, and Lyman 1987) is never mentioned. It is also claimed that the system's virtues "arise naturally from an approach that roughly corresponds to models of human natural language processing" (p. 160), although there is no consideration of any psycholinguistic models. Webber's (1978) work on verb-phrase ellipsis is not discussed (though there is passing mention of her formal approach to discourse representation); Frederking apparently rejects Webber's conclusions as to what representations and processes are required in VP ellipsis resolution but doesn't say why.

The content of the book is Frederking's Ph.D. thesis from Carnegie Mellon University, with only the addition of an index and a change of title. (In the thesis title, "integrated" qualified "model", not "dialogue"; the significance of the change is unclear.) Only a small part of the book (20 pages) discusses theoretical aspects of the work; the bulk concerns details of the implementation in OPS5 (58 pages), and long annotated traces of the program (54 pages), with introduction, literature review, and conclusions taking the remaining space. Thus conference or journal publication might have been more