Foundational Issues in Natural Language Processing

edited by Peter Sells, Stuart M. Shieber, and Thomas Wasow

System Development Foundation Benchmark Series

A Bradford Book
The MIT Press
Cambridge, Massachusetts
London, England

This book was set in Palatino by Asco Trade Typesetting Ltd., Hong Kong and printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Foundational issues in natural language processing / edited by Peter Sells, Stuart M.
  Shieber, and Thomas Wasow.
  p. cm.—(System Development Foundation benchmark series)
  "A Bradford book."
  Includes bibliographical references and index.
  ISBN 0-262-19303-5
  1. Computational linguistics. I. Sells, Peter, 1957– . II. Shieber, Stuart M.
III. Wasow, Thomas. IV. Series.
P98.F65 1991
410'.285—dc20 91-599
CIP

Foundational Issues in Natural Language Processing

System Development Foundation Benchmark Series

Max Brady, editor
Robotics Science, 1989

Max V. Mathews and John R. Pierce, editors
Current Directions in Computer Music Research, 1989

Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors
Intentions in Communication, 1990

Eric L. Schwartz, editor
Computational Neuroscience, 1990

Peter Sells, Stuart M. Shieber, and Thomas Wasow, editors
Foundational Issues in Natural Language Processing, 1991

List of Contributors

Robert C. Berwick
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA

Janet Dean Fodor
Graduate Center
City University of New York
New York, NY

Aravind K. Joshi
Department of Computer and Information Science
Moore School
University of Pennsylvania
Philadelphia, PA

William C. Rounds
Computer Science and Engineering Division
Electrical Engineering and Computer Science Department
University of Michigan
Ann Arbor, MI

K. Vijay-Shanker
Department of Computer Science
University of Delaware
Newark, DE

David Weir
Department of Electrical Engineering and Computer Science
The Technological Institute
Northwestern University
Evanston, IL

Contents

List of Contributors vii

Introduction 1

Chapter 1
The Relevance of Computational Complexity Theory to Natural Language Processing 9
William C. Rounds

Chapter 2
The Convergence of Mildly Context-Sensitive Grammar Formalisms 31
Aravind K. Joshi, K. Vijay-Shanker, and David Weir

Chapter 3
Sentence Processing and the Mental Grammar 83
Janet Dean Fodor

Chapter 4
Principle-Based Parsing 115
Robert C. Berwick

Index 227

ntroduction

Research on natural language processing involves at least four distinct but closely related areas of study. They are (1) investigating the psychological processes involved in human language understanding; (2) building computational systems for analyzing natural language input (and/or producing natural language output); (3) developing theories of natural language structure; and (4) determining the mathematical properties of grammar formalisms.

(1) and (2) are obvious lines of research, differing in their emphasis on whether the goal is to understand human language use or to simulate human linguistic behavior on a machine. In order for these efforts to have some principled basis, they must incorporate (or perhaps embody) theoretical claims about the appropriate units of description for utterances and what relations among those units are linguistically significant. In other words, significant work in areas (1) and (2) builds on (and, ideally, contributes to) work in area (3). In all three of these lines of research, there is a tendency to develop rich systems of formalism and terminology. Area (4) plays a vital clarifying role in the enterprise, by specifying rigorously what the relations are among different-looking models of natural language structure and processing.

The literature abounds with examples of work that demonstrates the interrelatedness of these areas of research, bringing considerations from one to bear on the problems of another. A particularly clear illustration is provided by Chomsky's argument in the 1950s (see Chomsky 1956, 1957) that finite-state grammars were inadequate models of natural language syntax. Let us review that argument, highlighting the roles of (1)–(4) and the relations among them (and, in the process, perhaps stretching the historical record just a little).

A *finite-state grammar* is a rewriting system, with rules of the following forms (where capital letters are nonterminal symbols and lowercase letters are terminal symbols): $A \rightarrow a$ and $A \rightarrow Ba$. The language generated by such a grammar is the set of strings of terminal symbols that can be generated beginning with the designated nonterminal symbol S and applying rules from the grammar in any order. Such a language is called a *finite-state* *language*. It is evident that every finite-state grammar is equivalent to an automaton, described as follows by Chomsky (1957 : 18–19):

> Suppose that we have a machine that can be in any one of a finite number of different internal states, and suppose that this machine switches from one state to another by producing a certain symbol (let us say, an English word). One of these states is an initial state; another is a final state. Suppose that the machine begins in the initial state, runs through a sequence of states (producing a word with each transition), and ends in the final state. Then we call the sequence of words that has been produced a "sentence". Each such machine thus defines a certain language; namely, the set of sentences that can be produced in this way.

Chomsky (1957 : 20) goes on to say:

> This conception of language is an extremely powerful and general one. If we can adopt it, we can view the speaker as being essentially a machine of the type considered. In producing a sentence, the speaker begins in the initial state, produces the first word of the sentence, thereby switching into a second state which limits the choice of the second word, etc. Each state through which he passes represents the grammatical restrictions that limit the choice of the next word at this point in the utterance.

What we have, then, is a very simple theory of grammar, a corresponding machine model that could easily be implemented on real computers (even those of the mid-1950s), and the suggestion that this model might be given a psychological interpretation. Simple as it is, this approach to natural language processing was not made entirely of straw. Chomsky (1957 : 20) says that this "is essentially the model of language" developed in Hockett 1955, and it is possible to find statements in the psychological literature of the time to indicate that it was taken seriously as an account of human verbal behavior. For example, Lashley (1951 : 182) quotes the following characterization of language behavior from Washburn 1916: "a combination of movements so linked together that the stimulus furnished by the actual performance of certain movements is required to bring about other movements."

It is not difficult to prove that certain sets of strings are not finite-state languages. For example, "mirror-image languages"—that is, infinite sets of strings, each of which is a palindrome (such as {*a, b, aa, bb, aaa, bbb, aba, bab, aaaa, bbbb, abba, baab, . . .*})—are not finite state. More generally, any language whose sentences can have an unbounded number of nested dependencies is beyond the expressive power of finite-state grammars or the equivalent machines. The heart of Chomsky's argument against finite-state grammars was the claim that natural languages permit precisely this sort of dependency.

English, he observed (Chomsky 1957:22), includes constructions like those in (1), where the *S*'s represent clauses.

(1) a. If S_1, then S_2.
b. Either S_1 or S_2.

Other combinations, such as **Either* S_1 *then* S_2, are not possible, showing that there is a dependency between the words in each of these pairs. Moreover, sentences of the forms in (1) can be substituted for the *S*'s in these schemata, yielding patterns like *If either* S_1 *or* S_2, *then* S_3. In principle, this process can be iterated, resulting in strings of arbitrary length with unbounded numbers of these dependent elements, nested in just the manner known to be beyond the descriptive power of finite-state grammars.

We note in passing that neither Chomsky nor we present actual English sentences (as opposed to schematic patterns) in support of this argument. We return to the significance of this omission below. For now, we wish to emphasize the form of the argument: a mathematical result and an observation about the well-formed sentences of English are combined to discredit a theory of grammar and models of natural language processing (both psychological and computational) built on that theory. This is a prototype for arguments connecting mathematical linguistics, grammatical theory, psycholinguistics, and computational linguistics.

The elegance of Chomsky's argument led others to seek similar results. Few, if any, have been as celebrated, in part because developing comparably powerful arguments for or against less simplistic theories proved more difficult. However, the literature contains many examples, of which we will briefly describe four.

Putnam (1961) argued that natural languages (assumed to be sets of sentences, which in turn were taken to be strings of words) must be decidable. This was based on the observation that people are very good at distinguishing well-formed sentences from arbitrary strings of words. Since the human brain is, according to Putnam, a finite computing device, it follows that there must be an effective procedure for each language capable of deciding which strings are sentences of the language. Putnam coupled this with an argument that the theory of transformational grammar prevalent at the time allowed grammars that would generate undecidable languages. The latter argument was based on the observation that transformational grammars could mimic arbitrary Turing machine operations by means of insertions and deletions. Hence, Putnam concluded, transformational grammar as it was being developed at the time was too powerful a theory of natural language syntax. Here mathematical and psychological considerations were combined to argue against a linguistic theory.

Postal emulated Chomsky's argument more directly. Citing results (Chomsky 1959) showing that context-free grammars could not generate languages with arbitrary cross-serial dependencies—that is, dependencies of the form $a_1a_2a_3 \ldots a_n \ldots b_1b_2b_3 \ldots b_n \ldots$ (where the dependencies hold between the a's and b's with the same subscripts)—Postal (1964a) claimed that the Northern Iroquoian language Mohawk had a construction of precisely this form. Postal (1964b) went on to argue that a multitude of then popular grammatical theories were in fact merely unformalized versions of context-free grammar. He did not go on to draw the associated psychological and computational inferences, though others did (see, for example, Chomsky 1964: sec. I; Hopcroft and Ullman 1979:78).

Gazdar (1981:155) argued on the basis of psycholinguistic and computational considerations *for* taking context-free grammar seriously as a theory of natural language syntax. The fact that context-free languages can be parsed in time (at worst) proportional to the cube of the length of the input (Earley 1970) comports well with the observation that humans are very efficient at processing natural language input, as well as with the aim of constructing fast natural language understanding systems. These mathematical, psychological, and computational observations served as initial motivation for the grammatical theory known as Generalized Phrase Structure Grammar (see Gazdar et al. 1985 for a detailed exposition of this theory, as well as arguments for it based on linguistic considerations).

A somewhat different combination of mathematical, psychological, and linguistic considerations can be found in work on learnability, notably the work of Wexler and Culicover (1980). Starting from the observation that natural languages are learned by young children, they developed a precise definition of learnability. They went on to propose a set of constraints on a version of transformational grammar that would jointly suffice to permit a proof that the languages such grammars generate were learnable—in fact, learnable from simple data.

Other examples could be cited. The point, however, should by now be clear: the four areas of research cited in the opening paragraph have often been related to one another in the literature. On the other hand, such connections have also often been questioned, and inferences like those described in the preceding paragraphs are regarded by many with considerable skepticism. A number of factors contribute to the difficulty of making persuasive arguments relating psycholinguistics, computational linguistics, grammatical theory, and mathematical linguistics.

First, and most widely discussed, is the fact that a grammar does not determine a processing system (either human or electronic). The performance of a language processor depends on many factors other than the knowledge embodied in a grammar. Observable behaviors—including both grammaticality judgments and measures of processing complexity like reaction times—are influenced by such extragrammatical factors as memory limitations, parsing strategies, lexical retrieval mechanisms, and the like. Hence, any argument purporting to connect data on language processing to grammatical theory is open to serious question.[1]

Indeed, it is not even clear that an optimal processing system should include a discrete part that can be identified as the grammar. Much work has assumed that it should; Chomsky (1965 : 9), for example, states, "No doubt, a reasonable model of language use will incorporate, as a basic component, the generative grammar that expresses the speaker-hearer's knowledge of the language." This assumption has been questioned on a number of occasions (see, for example, Bever 1970; Katz 1981), and the issue remains a vexed one. It is addressed in the present volume in the chapters by Berwick and Fodor.

Second, the appropriate notions of complexity for evaluating natural language processing systems are not given a priori. Chomsky's argument against finite-state grammar reviewed earlier takes generative capacity as its complexity measure, but this is a very crude metric. For example, though deterministic and nondeterministic finite-state machines accept precisely the same languages, a deterministic machine may require exponentially more states than a nondeterministic one for a given language.

In the past decades, theoretical computer science has provided a variety of complexity measures more delicate than generative capacity. However, these measures may depend crucially on the form of the system. Hence, arguments relating grammatical theory to processing can make use of these measures only if they make strong assumptions about how the grammar is embedded in the processing system. For example, Berwick and his collaborators have shown that generalized phrase structure grammars, although equivalent in generative capacity to context-free grammars, fare far worse with respect to some other complexity metrics, thus weakening considerably the force of Gazdar's argument cited above. Rounds's chapter in the present volume surveys some of the mathematical tools available for the comparison of natural language systems, as well as some of the ways in which care must be taken in their application.

Third, the mathematical results employed in arguments of the sort we are considering tend to be worst-case, limiting arguments. Familiar context-free languages, for example, can be parsed in linear time—far faster than the cubic time worst-case result available for the class as a whole. The mathematical results, therefore, may not always be a useful guide to the performance of real processing systems.

The issue comes up even in connection with Chomsky's argument against finite-state grammars. That argument depends crucially on the grammaticality of sentences with arbitrarily many nested dependencies. In fact, speakers tend to have great difficulty with sentences containing more than two nested dependencies. For example, even a sentence like (2) is quite awkward, though it was carefully constructed to exhibit three nested dependencies without becoming utterly incomprehensible.

(2) If Pat both sees either Chris or Sandy and talks to one of them, then we should leave.

Further nesting would render it completely unacceptable. Moreover, one could argue that, because every human has only a finite amount of memory, natural languages *must* be finite state, for any language accepted by an automaton with fixed finite memory can be shown to be finite state. Consequently, Chomsky's argument (and most others modeled on it) requires that we dismiss as irrelevant factors like memory limitations. Since such factors are very relevant to the performance of real processing systems, the significance of the limiting proofs is called into question.

In short, the connections among grammatical theory, mathematical linguistics, and the operation of real natural language processing systems are complex ones. Drawing conclusions in one domain on the basis of considerations from another typically involves making simplifying assumptions that are easily called into question. This does not mean that the four lines of inquiry in question should proceed completely independently of one another. If the construction of computational natural language systems is to be of scientific as well as practical significance, it must be related in some way to human linguistic processing. And if the latter is to be studied in a systematic way, experiments must be based on a rigorous and principled theoretical foundation. In spite of the difficulties inherent in trying to relate the formal properties of grammatical theories to the observable properties of language-processing systems, the alternative to facing up to these difficulties is mere seat-of-the-pants guesswork. For this reason, work continues on the relationship among these four areas, and much of the best research on natural language processing—including the research presented in this volume—concerns itself with those relationships.

The chapters by Rounds and by Joshi, Vijay-Shanker, and Weir deal with the relationship of mathematical results about grammar formalisms to linguistic issues. Rounds discusses the relevance of complexity results to linguistics and computational linguistics, providing useful caveats about how results might be misinterpreted, plus pointers to promising avenues of future research. Joshi, Vijay-Shanker, and Weir survey results showing the equivalence (in terms of generative capacity) of several different grammatical formalisms, all of which are "mildly context sensitive." Central to their results are a number of variants on tree-adjoining grammars, a formalism they and several collaborators have developed and applied to natural language phenomena.

The chapter by Fodor is concerned with the relationship of grammatical or computational models to psychological processes in the minds of speakers. Fodor discusses how psycholinguistic results can bear on the choice among competing grammatical theories, surveying a number of recent experiments and their relevance to issues in grammatical theory.

The chapter by Berwick considers the relationship between issues in linguistic theory and the construction of computational parsing systems. Berwick examines what it means to implement a theory of grammar in a computational system. He argues for the advantages of a "principle-based" approach over a "rule-based" one and surveys several recent parsing systems based on Government-Binding Theory.

The four chapters contain revised versions of material presented at a conference held in January, 1987, in Santa Cruz, California. The conference was sponsored by the Center for the Study of Language and Information, with funds provided by the System Development Foundation. In addition to the four papers published here, there were presentations by Lauri Karttunen and Don Hindle. Karttunen explored unification-based approaches to grammatical analysis and their appeal from both linguistic and computational perspectives. Hindle reported on his work with Mitch Marcus on the computational and psychological advantages of deterministic parsing models.

The issues addressed in this volume are difficult ones. They will continue to be debated for decades to come. These works represent the current state of the art as articulated by some of the leading thinkers in the multidisciplinary field of natural language processing.

Note

1. This argument applies to all kinds of performance data, including native speakers' judgments of acceptability. Although it is standard for generative grammarians to take such judgments as providing especially direct access to some internalized grammar, no justification for this practice has ever been offered.

References

Bever, T. 1970. "The Cognitive Basis for Linguistic Structures." In J. R. Hayes (ed.) *Cognition and Language Learning*. New York: Wiley.

Chomsky, N. 1956. "Three Models for the Description of Language." *I.R.E. Transactions on Information Theory* IT-2:113–124. Reprinted, with corrections, in R. Luce, R. Bush, and E. Galanter (eds.) *Handbook of Mathematical Psychology, Volume II*. New York: Wiley (1965).

Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton.

Chomsky, N. 1959. "On Certain Formal Properties of Grammars." *Information and Control* 2:137–167. Reprinted in R. Luce, R. Bush, and E. Galanter (eds.) *Handbook of Mathematical Psychology, Volume II*. New York: Wiley (1965).

Chomsky, N. 1964. "Current Issues in Linguistic Theory." In J. A. Fodor and J. Katz (eds.) *The Structure of Language*. Englewood Cliffs, New Jersey: Prentice-Hall.

Chomsky, N. 1965. *Aspects of the Theory of Syntax*. Cambridge, Massachusetts: MIT Press.

Earley, J. 1970. "An Efficient Context-free Parsing Algorithm." *Communications of the ACM* 13:94–102.

Gazdar, G. 1981. "Unbounded Dependencies and Coordinate Structure." *Linguistic Inquiry* 12:155–184.

Gazdar, G., E. Klein, G. Pullum, and I. Sag. 1985. *Generalized Phrase Structure Grammar.* Oxford: Blackwell.

Hockett, C. 1955. *A Manual of Phonology.* Baltimore: Waverly Press.

Hopcroft, J., and J. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation.* Reading, Massachusetts: Addison-Wesley.

Katz, J. 1981. *Language and Other Abstract Objects.* Totowa, New Jersey: Rowman and Littlefield.

Lashley, K. 1951. "The Problem of Serial Order in Behavior." In S. Saporta (ed.) *Psycholinguistics.* New York: Holt, Rinehart and Winston.

Postal, P. 1964a. "Limitations of Phrase Structure Grammar." In J. A. Fodor and J. Katz (eds.) *The Structure of Language.* Englewood Cliffs, New Jersey: Prentice-Hall.

Postal, P. 1964b. *Constituent Structure: A Study of Contemporary Models of Syntactic Description.* Bloomington, Indiana: Research Center for the Language Sciences.

Putnam, H. 1961. "Some Issues in the Theory of Grammar." In G. Harman (ed.) *On Noam Chomsky.* Garden City, New York: Anchor Books.

Washburn, M. F. 1916. *Movement and Mental Imagery.* Boston: Houghton Mifflin.

Wexler, K., and P. Culicover. 1980. *Formal Principles of Language Acquisition.* Cambridge, Massachusetts: MIT Press.

Chapter 1

The Relevance of Computational Complexity Theory to Natural Language Processing

William C. Rounds

1 *Introduction*

Mathematical models in linguistics often have a peripheral status. The creator of a model may not completely understand the theory being modeled, and the user of a model may not understand its idealizations and presuppositions. The result may be that the model is generally ignored, or that its predictions are used too literally. It is therefore necessary for the creators of models to document exactly their intentions for the models, and for the users of the models to be aware of these intentions and to realize what assumptions and idealizations are made. This chapter is an attempt to explain informally the intuitions behind complexity theory in computer science, with a view toward discovering in what ways the results of this theory may be used productively in computational linguistics and linguistics more generally. Several papers have recently appeared invoking complexity techniques and deriving complexity results for various linguistic theories. These (generally excellent) papers deserve to have their presuppositions carefully examined, so that their conclusions may be properly applied.

One of the first formalizations of a linguistic theory was the paper by Peters and Ritchie (1973) on the generative capacity of transformational grammars. Their model is intended to capture the intuitions of transformational grammar as exactly as possible, given the extensive literature available at the time, and focusing principally on the *Aspects* model (Chomsky 1965). Peters and Ritchie were especially careful to document their intentions in making this model, with the result that its conclusions (that any recursively enumerable set could be generated) forced a major reassessment of the notion of "natural language grammar."

Complexity results in linguistics usually involve several parameters and often speak about more than simple notions of weak generative capacity. The problems of documentation are thus more difficult than those faced by Peters and Ritchie. However, identifying the right complexity parameters can still clarify some real issues for a linguistic theory. The papers by Ristad (1986a) and Barton (1985, 1986) are good examples. Ristad's results point to aspects of generalized phrase structure theory (Gazdar et al. 1985) that may lead to computational difficulties, and Barton's results point out similar problems in immediate dominance/linear precedence (ID/LP) parsing and in morphology.

These results have been summarized by Barton, Berwick, and Ristad (1987). I recommend that the reader consult this book for a full explanation of the notions I touch on in this chapter. Many of the points I make here are reiterated there.

The presuppositions of an already established theory, such as complexity theory, are perhaps the properties of the theory most easily ignored in making an application. The theory in this case is an attempt to classify decision problems in terms of the computational resources required by an abstract sequential machine. These assumptions need to be rephrased linguistically in order to apply the results sensibly. One needs to make a hypothesis that the brain is some sort of sequential computer, and that natural languages are infinite sets of strings, for example. It is worth noting that even in computer science, the presuppositions about abstract sequential machines have been challenged. Computers need not be sequential machines of potentially infinite capacity like Turing machines, but can be modeled as families of Boolean circuits or, more generally, as families of finite-capacity machines (Gurevich 1988). Incidentally, this fact suggests that connectionist models in linguistics need not be dismissed because they do not explain how infinite sets of strings can be generated. These models are quite closely related to the Boolean circuit models in computer science, and it would seem that a synthesis of ideas is possible here.

In the light of these remarks, I have decided to examine several areas of complexity theory and to include for each area an analysis of its presuppositions. I will also include for each area an example of application (or misapplication) of its results and will try to give an intuitive feeling for the relevance of the results in linguistics. (In addition to the book mentioned above, I should note Berwick's book (1985), the book by Berwick and Weinberg (1984), and Perrault's survey article (1984) as other good contributions in the same spirit.) In the concluding section I will suggest some ways in which complexity techniques may lead to the discovery of new linguistic properties, or to new ways of regarding old phenomena. As a tentative example, I will focus on a model for studying language learnability, based on techniques from complexity theory and adapted from a very interesting model of Valiant (1984). This adaptation may well require further changes, but this should be true of all formal models. They need not be frozen in the realm of abstractions but should, when necessary, be reshaped to fit empirical data and to explain new linguistic hypotheses. Doing so calls for pooling the expertise of linguists, computer scientists, and mathematicians in a truly collaborative effort.