A System for Natural Language Sentence Generation

Michael Levison and Gregory Lessard

Department of Computing and Information Science, Queen's University, Kingston, Ontario, Canada, K7L 3N6
e-mail: levison@qucis.queensu.ca

and

Department of French Studies, Queen's University, Kingston, Ontario, Canada, K7L 3N6
e-mail: lessard@francais.queensu.ca

Abstract: This paper describes a natural language generation system known as VINCI, which accepts as input a formal description of some subset of a natural language, and generates strings in the language. With the help of an attribute grammar formalism, the system can be used to simulate on a computer components of several current linguistic theories. The program, implemented in C, runs under a variety of operating systems, including UNIX, MS-DOS and VM/CMS. In this paper we consider not only the design of the system, but also some of its applications in linguistic modelling and second language acquisition research.

Key Words: natural language generation, linguistic modelling, attribute grammar, applied linguistics, language learning.

Michael Levison is professor and Head of the Department of Computing and Information Science at Queen's University at Kingston. His research interests include natural language generation and editing systems. He has written many articles on computer applications in literary studies, and is a co-author of The Settlement of Polynesia: A Computer Simulation *(University of Minnesota Press, 1973).*

Gregory Lessard is an associate professor in the French Studies Department at Queen's University at Kingston. His research interests include natural language generation and modelling, second language acquisition and computer-aided text analysis.

Computers and the Humanities **26**: 43—58, 1992.

1. Introduction

1.1. *Background*

Most current linguistic theories presuppose the notion that human language is describable by a finite set of rules capable of generating an infinite set of sentences. Thus, in theory, the goal of linguistic research is to define for each language a set of rules (the grammar) which generates all sentences considered by native speakers to be grammatical, and none considered to be ungrammatical. At a more general level, its goal is to characterize human language in general. In practice, no complete generative grammar exists for any language, and linguists tend to focus on small subsets of a language and on specific linguistic levels (syntax or morphology or semantics), and to propose often relatively informal rules which they test against their own linguistic intuition, and that of their colleagues.

This approach is not without its weaknesses. Linguistic intuition is not an absolutely reliable source of data (Birdsong, 1989). Not only is considerable variation to be found among (and even within) subjects, but the devices used to access metalinguistic judgements are themselves not immune to the influence of extraneous factors such as pragmatic and contextual variation.

An appeal to linguistic corpora as a richer source of data (Aarts and Meijs, 1984) goes some way toward resolving this dilemma, particularly given the growing size and number of available corpora and the increasing sophistication of text analysis software. However, while corpus data illustrate a range of utterances, and force us to go beyond our own intuitions, they rarely provide insight into what may not be produced as an utterance, except in a limited, statistical sense (structure x is not found, while structure y appears z times).

For all these reasons, we believe that it is important to complement intuition both with corpus data and with computer-aided modelling based on grammars postulated by the linguist.

VINCI is a program which accepts as input a formal description of a natural language, and generates strings in the language. It should be made clear from the outset, however, that it is not our goal in the VINCI project to propose a grammatical formalism to compete with current models such as Government and Binding (Chomsky, 1981), Lexical Functional Grammar (Kaplan and Bresnan, 1982), or Generalized Phrase Structure Grammar (Gazdar et alii, 1985), or any other model. Indeed, given that the aim of current linguistic theory consists in attempting to model human language with a minimum of formal devices (as Gazdar et alii point out, for example, in rejecting transformations), the system described here far outstrips the needs of current linguistic theory.

Our goal is rather to provide an implementation capable of allowing the linguist to embody some minimum number of constructs from his or her favourite theory. It is worth emphasizing that theoretical constraints are defined by the model proposed by the linguist, and not by the implementation environment.

The VINCI system is thus at the interface between two currents in natural language generation research. On the one hand, many projects have as their primary goal the development of a workable system designed to respond to a particular set of needs. For example, researchers such as McKeown (1985) and Hovy (1991) have been concerned with the production of coherent text. Others, like Danlos have dealt with the question of lexical selection (Danlos, 1987), while still others have worked on problems of user modelling and reactive feedback (Moore & Swartout, 1991).[1]

At the other end of the spectrum, a smaller number of projects have had as their goal the computational implementation of a particular linguistic theory, quite apart from its applications. Thus, Friedman (1972) proposed a generative environment based on the then-current Transformational-Generative paradigm. More recently, Patten & Ritchie (1987) have produced an implementation of Systemic Grammar, Bresnan et alii (1987) have produced a parser for testing Lexical Functional Grammar rules, and Dahl (1989) has described an implementation of the Discontinuous Grammar formalism.

It can be seen that VINCI shares elements of both approaches. On the one hand, it is an attempt to provide an environment for solving a particular range of problems, among them teaching basic concepts of linguistic theory, modelling performance errors and providing a generative computer-aided language learning environment. On the other hand, it attempts to provide the most common devices found in mainstream generative grammar. In this paper, we shall discuss some principles which underly the system, the system itself, and some of its applications.

1.2. *Some principles*

Several guiding principles have strongly influenced the characteristics of the current system.

Firstly, in our opinion, the user of the system, typically a linguist or other language specialist, should be allowed to express grammatical rules in a formalism with which he or she is familiar, rather than learning to program in PROLOG, or something similar.[2] To this end VINCI provides the linguist with metalanguages as similar as possible to those used in current linguistic research.

Secondly, we believe that a linguist wishing to model a phenomenon at one particular level, say syntax, should not have first to construct a morphology and a lexicon in order to obtain evaluable output. While it is not our intent to construct a "complete" grammar for any language, we do envisage that libraries of predefined linguistic data and rules will gradually be assembled, which can be called upon where needed, to make it as easy as possible for a user to explore a particular language subset. For example, in a recent project which used the VINCI system (Lessard, Levison and Olson, 1991), a linguist