Syntactic Processing

Martin Kay
Xerox Palo Alto Research Center

In computational linguistics, which began in the 1950's with machine translation, systems that are based mainly on the lexicon have a longer tradition than anything else---for these purposes, twenty five years must be allowed to count as a tradition. The bulk of many of the early translation systems was made up by a dictionary whose entries consisted of arbitrary instructions in machine language. In the early 60's, computational linguists---at least those with theoretical pretentions---abandoned this way of doing business for at least three related reasons:

First systems containing large amounts of unrestricted machine code fly in the face of all principles of good programming practice. The syntax of the language in which linguistic facts are stated is so remote from their semantics that the opportunities for error are very great and no assumptions can be made about the effects on the system of invoking the code associated with any given word. The systems became virtually unmaintainable and eventually fell under their own weight. Furthermore, these failings were magnified as soon as the attempt was made to impose more structure on the overall system. A general backtracking scheme, for example, could all too easily be thrown into complete disarray by an instruction in a single dictionary entry that affected the control stack.

Second, the power of general, and particularly nondeterministic, algorithms in syntactic analysis came to be appreciated, if not overappreciated. Suddenly, it was no longer necessary to seek local criteria on which to ensure the correctness of individual decisions made by the program provided they were covered by more global criteria. Separation of program and linguistic data became an overriding principle and, since it was most readily applied to syntactic rules, these became the main focus of attention.

The third, and doubtless the most important, reason for the change was that syntactic theories in which a grammar was seen as consisting of a set of rules, preferably including transformational rules, captured the imagination of the most influential noncomputational linguists, and computational linguists followed suite if only to maintain theoretical respectability. In short, systems with small sets of rules in a constrained formalism and simple lexical entries apparently made for simpler, cleaner, and more powerful programs while setting the whole enterprise on a sounder theoretical footing.

The trend is now in the opposite direction. There has been a shift of emphasis away from highly structured systems of complex rules as the principle repository of information about the syntax of a language towards a view in which the responsibility is distributed among the lexicon, semantic parts of the linguistic description, and a cognitive or strategic component. Concomitantly, interest has shifted from algorithms for syntactic analysis and generation, in which the control structure and the exact sequence of events are paramount, to systems in which a heavier burden is carried by the data structure and in which the order of events is a matter of strategy. This new trend is a common thread running through several of the papers in this section.

Various techniques for syntactic analysis, notably those based on some form of Augmented Transition Network (ATN), represent grammatical facts in terms of executable machine code. The dangers to which this exposed the earlier systems are avoided by insisting that this code by compiled from statements in a formalism that allows only for linguistically motivated operations on carefully controlled parts of certain data structures.

The value of nondeterministic procedures is undiminished, but it has become clear that it does not rest on complex control structures and a rigidly determined sequence of events. In discussing the syntactic processors that we have developed, for example, Ron Kaplan and I no longer find it useful to talk in terms of a parsing algorithm. There are two central data structures, a chart and an agenda. When additions to the chart give rise to certain kinds of configurations in which some element contains executable code, a task is created and placed on the agenda. Tasks are removed from the agenda and executed in an order determined by strategic considerations which constitute part of the linguistic theory. Strategy can determine only the order in which alternative analyses are produced. Many traditional distinctions, such as that between top-down and bottom-up processing, no longer apply to the procedure as a whole but only to particular strategies or their parts.

This looser organization of programs for syntactic processing came, at least in part, from a generally felt need to break down the boundaries that had traditionally separated morphological, syntactic, and semantic processes. Research directed towards speech understanding systems was quite unable to respect these boundaries because, in the face of uncertain data, local moves in the analysis on one level required confirmation from other levels so that a common data structure for all levels of analysis and a schedule that could change continually were of the essence. Futhermore, there was a mouvement from within the artificial-intelligence community to eliminate the boundaries because, from that perspective, they lacked sufficient theoretical justification.

In speech research in particular, and artificial intelligence in general, the lexicon took on an important position if only because it is there that the units of meaning reside. Recent proposals in linguistic theory involve a larger role for the lexicon. Bresnan (1978) has argued persuasively that the full mechanism of transformational rules can, and should, be dispensed with except in cases of unbounded movement such as relativization and topicalization. The remaining members of the familiar list of transformations can be handled by weaker devices in the lexicon and, since they all turn out to be lexically governed, this is the appropriate place to state the information.

Against this background, the papers that follow, different though they are in many ways, constitute a fairly coherent set. Carbonell comes from the artificial-intelligence tradition and is generally concerned with the meanings of words and the ways in which they are collected to give the meanings of custom-built pieces of discourse. In the present paper, he explores ways in which this process can be made to reflect back on itself to fill gaps in the lexicon by appropriate analysis of the context. At its base, the method is familiar from similar work in syntax. The missing element is treated as though it had whatever properties allow a coherent analysis of the larger unit---say a sentence, or paragraph---in which it is embedded. These properties are then entered against it in the lexicon for future use. The problem, which is faced in this paper, is that the possibility that the lexicon is deficient must be faced in respect of all words because, even when there is an entry in the lexicon, it may not supply the reading required in the case on hand. Small, like Carbonell is concerned with the meanings of words and he is lead to a view of words as active agents. The main role of the system is to act as moderator.

Kwasny and Sonheimer have a concern to Carbonell's. When problems arise in analysis, they look for deficiencies in the text rather than in the lexicon and the rules. It is no indictment of either paper that they provide no way of distinguishing the cases, for this is clearly a separate enterprise. Kwasny and Sonheimer propose progressively weakening the requirements that their analysis system makes of a segment of text so that, if it does not accord with the best principals of composition, an analysis can still be found by taking a less demanding view of it. Such a technique clearly rests on a regime in which the scheduling of events is relatively free and the control structure relatively free.

Shapiro shows how a strong data structure and a weak control structure make it possible to extend the ATN beyond the analysis of one dimensional strings to semantic networks. The result is a total system with remarkable consistency in the methods applied at all levels and, presumably, corresponding simplicity and clarity in the architecture of the system as a whole.

Allen is one of the foremost contributors to research on speech understanding, and speech processing in general. He stresses the need for strongly interacting components at different levels of analysis and, to that extent, argues for the kind of data-directed methods I have tried to characterize.

At first reading, Eisenstadt's paper appears least willing to lie in my Procrustean bed, for it appears to be concerned with the finer points of algorithmic design and, to an extent, this is true. But, the two approaches to syntactic analysis that are compared turn out to be, in my terms, algorithmically weak. The most fundamental issues that are being discussed therefore turn out to concern what I have called the strategic component of linguistic theory, that is with the rules according to which atomic tasks in the analysis process are scheduled.

Reference

Bresnan, Joan (1978) "A Realistic Transformational Grammar" in Halle, Bresnan and Miller (eds.) Linguistic Theory and Psychological Reality, The MIT Press.