AI Magazine Volume 13 Number 1 (1992) (© AAAI)

A Flexible, Parallel Generator of Natural Language

Nigel Ward

My Ph.D. thesis (Ward 1992, 1991)[1] addressed the task of generating natural language utterances. It was motivated by two difficulties in scaling up existing generators. Current generators only accept input that are relatively poor in information, such as feature structures or lists of propositions; they are unable to deal with input rich in information, as one might expect from, for example, an expert system with a complete model of its domain or a natural language understander with good inference ability. Current generators also have a very restricted knowledge of language—indeed, they succeed largely because they have few syntactic or lexical options available (McDonald 1987)—and they are unable to cope with more knowledge because they deal with interactions among the various possible choices only as special cases.

To address these and other issues, I built a system called FIG (flexible incremental generator). FIG is based on a single associative network that encodes lexical knowledge, syntactic knowledge, and world knowledge. Computation is done by spreading activation across the network, supplemented with a small amount of symbolic processing. Thus, FIG is a spreading activation or structured connectionist system (Feldman et al. 1988).

In the initial state, some nodes representing concepts are sources of activation; this pattern of activation represents the information to be expressed. Activation flows from these nodes to nodes representing words through the various knowledge structures of the network. When the network settles, the most highly activated word is selected and emitted. Activation levels are then updated to represent the new current state, both in syntactic and semantic aspects. This process of settle, emit, and update repeats until all the input has been conveyed. An utterance is simply the result of successive word choices.

The treatment of syntax in connectionist and spreading activation systems is a well-known problem. In FIG, syntactic knowledge is encoded with network structures representing constructions and their constituents, drawing on some ideas of construction grammar (Fillmore 1988). Constituents are linked to words, syntactic categories, relations, and other constructions. Activation flow through these links and, eventually, to words provides for constituency and subcategorization. For a word to receive enough activation to be selected, it typically must receive activation from both constructions and the input, directly or indirectly; thus, each word choice is appropriate syntactically and semantically. The links to constituents are gated by cursors, which are updated over time based on feedback from the word's output. This mechanism ensures that words and concepts that are syntactically appropriate become highly activated at the right time, which causes words and concepts to appear in the right order. For example, activation from common nouns through the determination construction results in articles becoming highly activated; thus, FIG produces "a peach" instead of "peach" or "peach a." Constructions activate not only syntactic categories but also relations; for example, the subject-predicate construction's first constituent is linked to the relation's causer, agent, and experiencer with various weights, which causes the activation of concepts appropriate for being expressed before the verb. Constructions operate in synergy; for example, in "John made the cake vanish," the word "make" receives activation from the periphrastic causative construction because it is appropriate for this meaning and from the second constituent of the subject-predicate construction because it is a verb.

> *FIG is based on a single associative network that encodes lexical…syntactic …and world knowledge.*

FIG's syntactic coverage is much broader than that of previous connectionist generators such as Gasser (1988); output include "once upon a time there lived an old man and an old woman," "one day the old woman went to a stream to wash clothes," and "John ate a peach with an old woman's fork." The success of this model in generating utterances of English and Japanese suggests that the complexity present in most treatments of syntax is unnecessary: FIG dispenses with the assembly of *syntactic structures*, constructions that affect the utterance only by the activation they transmit, directly or indirectly, to words. FIG does without a mechanism for explicit syntactic choice; any number of constructions are potentially active, competing or cooperating in parallel, and the choice among them is emergent. Phenomena traditionally considered to require instantiation and variable binding are handled in FIG with much simpler mechanisms. Grammatical output results not from constraints on the form of syntactic structures or the behavior of an algorithm but, rather, from the structure and weights of the network as a whole.

This paragraph summarizes the ways in which FIG addresses the issues that motivated its construction: It handles arbitrarily rich input because the number of nodes activated in the initial state makes no difference to its operation. It handles interaction among choices easily because it tends to settle into a state representing a compatible set of choices as a result of links among nodes that represent such choices. It handles trade-offs among competing goals without additional mechanism because all computation is in terms of numbers. Thus, FIG is the first generator potentially able to perform well at the complex generation tasks that will arise in the future. Of course, to realize this potential requires more experimentation with the details of activation flow and with ways to quickly extend the grammar and lexicon.

FIG also suggests a way to achieve more natural machine translation. In existing generators, the structure of the input closely governs that of the output (McDonald, Vaughan, and Pustejovsky 1987), and as a result, output are often awkward. FIG is more flexible; there is no direct mapping from its input to its output. Instead, word choice is simultaneously affected by semantic and syntactic considerations of the target language, making it better able to take advantage of idiosyncrasies of the language.

Abstracting from the implementation leads to several design principles; consideration of the nature of the generation task also shows that these principles are required for powerful generation. The principles include explicit representation of the current state and many uses of parallelism: *knowledge source parallelism*, the simultaneous activity of syntactic, lexical, and other considerations; *competitive parallelism*, the simultaneous activity of multitudes of alternative words and constructions; *part-wise parallelism*, the simultaneous consideration of words for all parts of the utterance; *evaluative parallelism*, the simultaneous evaluation of all sources of evidence for the relevance of a word, construction, or concept; and *synergistic parallelism*, the simultaneous activity of many constructions. Corroboration for these design principles is found in the fact that human speakers appear to manifest them, as evidenced by introspection and data on pauses, priming, and speech errors.

Acknowledgments

This work was supervised by Robert Wilensky and supported by the National Science Foundation through a graduate student fellowship; the Sloan Foundation through the Berkeley Cognitive Science Program; the Japanese Ministry of Education through a Kokuhi Shoogakukin; the Defense Advanced Research Projects Agency, monitored by the Space and Naval Warfare Systems Command, under contracts N00039-84-C-0089 and N00039-88-C-0292; and the Office of Naval Research under contract N00014-89-J-3205.

Note

[1] A copy of this dissertation is available from the Computer Science Division, University of California at Berkeley, Berkeley CA 94920. A revised and expanded version is to be published by Ablex (Ward 1992).

References

Feldman, J.; Fanty, M.; Goddard, N.; and Lynne, K. 1988. Computing with Structured Connectionist Networks. *Communications of the ACM* 31:170–187.

Fillmore, C. J. 1988. The Mechanisms of "Construction Grammar." In Proceedings of the Fourteenth Berkeley Linguistic Society Conference, 35–55. Berkeley, Calif.: Berkeley Linguistic Society.

Gasser, M. 1988. A Connectionist Model of Sentence Generation in a First and Second Language, Technical Report, UCLA-AI-88-13, Computer Science Dept., Univ. of California at Los Angeles.

McDonald, D. 1987. Natural Language Generation. In *Encyclopedia of Artificial Intelligence,* volume 1, ed. S. C. Shapiro, 642–655. New York: Wiley.

McDonald, D.; Vaughan, M.; and Pustejovsky, J. 1987. Factors Contributing to Efficiency in Natural Language Generation. In *Natural Language Generation*, ed. G. Kempen, 159–182. Dordrecht, The Netherlands: Nijhof.

Ward, N. 1992. *A Flexible, Parallel Generator of Natural Language*. Norwood, N.J.: Ablex. Forthcoming.

Ward, N. 1991. A Flexible, Parallel Generator of Natural Language. Ph.D. diss., Technical Report 91/629, Dept. of Computer Science, Univ. of California at Berkeley.

Nigel Ward is currently a post-doctorate student at the University of Tokyo, Sentanken (Research Center for Advanced Science and Technology). He tends to attack natural language processing problems by experimenting in Lisp, drawing inspiration from cognitive science and connectionism. His most recent project led to a feature-space representation of relational information as an alternative to deep-case relations and the like. He is currently exploring parallel approaches to syntax for parsing.