A Connectionist Language Generator

Nigel Ward
(University of Tokyo)

Norwood, NJ: Ablex Publishing Corporation (Ablex Series in Artificial Intelligence, edited by Yorick Wilks), 1994, xi+298 pp; hardbound, ISBN 0-89391-974-8, $55.00; paperbound, ISBN 1-56750-038-2, $27.50

Reviewed by
George Berg
University at Albany, SUNY

1. Introduction

This book, based upon Ward's Ph.D. thesis, describes his natural language generation system FIG. FIG uses a structured connectionist network to generate either English or Japanese output. The input to FIG comes either from a simple parser of Japanese, or test inputs created by hand. The output generated is a single sentence in either English or Japanese.

Ward adopts a machine-translation perspective, where many of the (over-) simplifying assumptions used in traditional natural language generation systems are inadequate. In particular, many generation systems use representations that are thinly disguised restatements of the desired natural language output. This trivializes generation and raises the question of how these representations are themselves created. Because of the differences in the languages, a generation system for machine translation must use general and rich representations of the concepts to be expressed. And the system must have the power and knowledge to convert these representations into an acceptable form in the output language. Starting from these and other assumptions, Ward developed his generator.

2. A Description of FIG

Except for some minor parts, FIG is implemented as a structured connectionist network. Concepts, syntactic constructs, lexical entries, etc., are all represented as nodes in this network. Each node has associated with it a numeric activation value. This value reflects the node's current importance in the network's processing. The effects that nodes have on one another are represented by weighted links between them. FIG works in discrete time steps, where the activation values of all the nodes are updated. The new activation value of a node is a function of its current activation value and the activation values of the units connected to it, as moderated by the weights on the connecting links. Thus, nodes can either increase or decrease the activation of related nodes.

FIG works as follows. Initially the concept nodes representing the desired output are activated. The activation spreads outward over the links from these nodes. This activates nodes representing other concepts, syntactic constructs, etc. related to the original nodes. As activation spreads through the network, the active nodes correspond to those elements necessary for the correct output of the first word. As the network settles into a stable state, the system will output the word corresponding to the lexical node with the highest activation. After this, the network updates its state and then resettles to output the second word. This process continues until the network has produced words satisfying the demands of all of its initially active concepts.

3. Philosophy and Design Decisions Underlying FIG

A connectionist network has only one very simple method of communication and control—activating and suppressing nodes by spreading numeric activation. Critically, such networks lack a central control mechanism; the only control and processing are the effects of the active nodes. Because of this, FIG has subnetworks to do all of the necessary processing in the distributed and parallel fashion necessary in a connectionist model. For instance, subnetworks are devoted to controlling which syntactic constructs are potentially applicable at a given point. These nodes and links interact with the active concept nodes to ensure that only appropriate lexical nodes become active as potential words to output. As another example, Ward uses nodes and links to implement a set of features that provide a general mechanism to output appropriate prepositions.

The reason that FIG uses the somewhat impoverished connectionist model is that it allows Ward to implement many of the features that he believes are necessary in a good generation system. Connectionism provides a naturally parallel model that is not overwhelmed by processing the rich inputs and knowledge necessary in his general generation system. Unlike systems where the construction of a syntactic structure drives the entire process, the syntactic subnetworks in FIG subtly guide the generation process, which is also influenced by conceptual, idiomatic, lexical, and other nonsyntactic information. No overt sentence structure is created during the generation. Because of this, FIG does not need to make premature syntactic choices; and thus no explicit mechanisms for backtracking in the case of a bad choice are necessary. In FIG, so long as multiple competing structures are compatible with the words output to this point, the model just operates with all of them active until the generation of a word forces a choice. At that time, the syntactic construct with the highest activations, and therefore the most influence, will prevail.

4. Who Should Read this Book?

Although structured connectionist models are not the current fashion in artificial-neural-network circles, it would be a disservice to Ward's work to dismiss it on this basis. Even the strongest partisan of distributed representations, or neural network haters for that matter, will find Ward's work thought-provoking. He questions many of the assumptions that we take for granted in our natural language processing work, and even if our own preferences are different from Ward's, we can benefit from his fresh perspective. I recommend this book to anyone who is interested in how a parallel generation system can work. I also recommend it to people interested in questioning standard generation models, and in intelligent alternatives that Ward has implemented.

5. Advantages and Disadvantages

The primary advantages of FIG are its parallelism and its flexibility. Connectionist networks are easily implemented in almost any parallel computer model in common use.

The lack of a centralized controller avoids many of the problems of synchronization and other bottlenecks that plague other sequential and parallel models.

As the value of large, useful systems becomes more appreciated in the computational linguistics community, the necessity of using machine learning with any nontrivial system becomes apparent. In theory, FIG is a flexible, extensible model. Adding to it does not require changing existing parts. Additional conceptual or lexical information, syntactic coverage, or even additional processing mechanisms to improve generation can be added by including the requisite nodes and links. These new parts of the network can then in principle work in cooperation and competition with the old parts of the network to generate text (but see below).

Two disadvantages of FIG come to mind, and both of these are related to extending the network. First, based on this reviewer's work on a similar connectionist parsing system, adding to the network can be a complex task. The exact nodes and the links and how they fit the existing network must be determined and tested. And, while more robust than might be expected, the values of the weights are critical. These values are sometimes finely balanced; a small change in one weight may significantly alter the behavior of the entire network. So, significant new parts of FIG cannot necessarily be simply "plugged in."

Second, adding learning to FIG may be difficult. Because it is a *structured* connectionist network, there is no easy way to use standard artificial-neural-network learning methods. Symbolic machine learning methods may work here, but much more needs to be known about the behavior of FIG when new components are added before learning can be effectively used. This may limit the practical usefulness of an otherwise very interesting and thought-provoking model of natural language generation.

George Berg is an associate professor in the Department of Computer Science and in the Program in Linguistics and Cognitive Science at the University at Albany, State University of New York. His research has included connectionist models of both generation and parsing. Berg's address is: Department of Computer Science, LI-67A, University at Albany, SUNY, Albany, NY 12222; e-mail: berg@cs.albany.edu