Subsymbolic Natural Language Processing

An Integrated Model of Scripts, Lexicon, and Memory

Risto Miikkulainen

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

This book was set in Palatino by Superscript Editorial Production Services (using ZzTEX) and was printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Miikkulainen, Risto.
Subsymbolic natural language processing : an integrated model of scripts, lexicon, and memory / Risto Miikkulainen.
p. cm. — (Neural network modeling and connectionism)
"A Bradford book."
Includes bibliographical references and indexes.
ISBN 0-262-13290-7
1. Neural networks (Computer science) 2. Natural language processing (Computer science) I. Title II. Series.
QA76.87.M54 1993
006.3'5—dc20 92-37285
CIP

Preface

Recently there has been a lot of excitement in cognitive science about the subsymbolic (i.e. parallel distributed processing, or distributed connectionist, or distributed neural network) approach. Subsymbolic systems seem to capture a number of intriguing properties of human-like information processing such as learning from examples, context sensitivity, generalization, robustness of behavior, and intuitive reasoning. These properties have been very difficult to model with traditional, symbolic techniques.

Within this new paradigm, the central issues are quite different (even incompatible) from the traditional issues in symbolic cognitive science, and the research has proceeded without much in common with the past. However, the ultimate goal is still the same: to understand how human cognition is put together. Even if cognitive science is being built on a new foundation, as can be argued, many of the results obtained through symbolic research are still valid, and could be used as a guideline for developing subsymbolic models of cognitive processes.

This is where DISCERN, the computer-simulated neural network model described in this book, fits in. DISCERN is built purely on parallel distributed mechanisms, but at the high level it consists of modules and information structures similar to those of symbolic systems, such as scripts, lexicon, and episodic memory. At the highest level of cognitive modeling, the symbolic and subsymbolic paradigms have to address the same basic issues. Outlining a parallel distributed approach to those issues is the purpose of this book. DISCERN is, above all, a prototype of a subsymbolic natural language processing system.

DISCERN was originally developed as a dissertation project, and I am indebted to several people who inspired and advised me in my graduate career. Very special thanks go to Michael Dyer for continuous support, many inspiring discussions, and excellent suggestions on earlier papers and drafts of the dissertation. I was also lucky to have a very helpful doctoral committee. Jacques Vidal initially pointed me to

feature maps, Josef Skrzypek provided constructive criticism on terminology, claims, and plausibility of the research, and Eric Halgren and Patricia Greenfield provided many pointers and both criticism and encouragement from the neuroscience and psychology point of view.

UCLA and especially the Artificial Intelligence Laboratory formed a stimulating environment for research. I learned a great deal from interactions with many people in the AI Lab, including Leo Belyaev, Charlie Dolan, Margot Flowers, Maria Fuenmayor, Mike Gasser, Jack Hodges, Trent Lange, Geunbae Lee, Valeriy Nenov, Mike Pazzani, Alex Quilici, Walter Read, John Reeves, Ron Sumida, Alan Wang, and Uri Zernik. Special thanks go to Trent Lange and Othar Hansson for useful comments on an earlier draft of this book and for good suggestions for naming the system (Trent originally came up with DISCERN; Othar is responsible for the best runner-up: SCript-HAcking NetworK).

DISCERN was built on previous work by several people, and many of them also provided valuable feedback during the course of the research. Discussions with Teuvo Kohonen and Jay McClelland led to many insights, as did suggestions by Jeff Elman and Dave Touretzky. Charlie Dolan, Robbie Jacobs, Trent Lange, Geunbae Lee, Dave Plaut, Mark St. John, and Ron Sumida kindly provided postscripts for the figures I have included in this book to describe their work. The merge clustering and principal component analysis program used in section 5.3.2 is due to Andreas Stolcke and Yoshiro Miyata.

The research was supported in part by Initial Teaching Alphabet (ITA) Foundation, the JTF program of the Department of Defense, W. M. Keck Foundation, and the Hewlett-Packard equipment grant to UCLA for artificial intelligence research, and in part by the Academy of Finland, the Finnish Cultural Foundation, Emil Aaltonen Foundation, Foundation for the Advancement of Technology, the Finnish Science Academy, Alfred Kordelin Foundation, Thanks to Scandinavia Foundation, Jenny and Antti Wihuri Foundation, and the Economic and Technological Sciences Foundation of Finland.

Contents

PART IV
Evaluation

PART I

Overview

Chapter 1
Introduction

1.1 *Task: Processing Script-Based Narratives*

DISCERN (DIstributed SCript processing and Episodic memoRy Network) is a distributed artificial neural network system[1] that learns to process simple stereotypical narratives. To see what DISCERN is up against, let us consider the following input examples:

(1) `John went to MaMaison. John asked the waiter for lobster. John left a big tip.`

(2) `John went to LAX. John checked in for a flight to JFK. The plane landed at JFK.`

(3) `John went to Radio-Shack. John asked the staff questions about CD-players. John chose the best CD-player.`

The first narrative mentions only three events about John's visit to MaMaison. Because restaurant visits are common experiences with very regular events, a human reader immediately assumes a number of events that certainly (or most likely) must have occurred. For example, he/she assumes that the waiter seated John, that John ate the lobster, and that John paid the waiter. Based on the fact that the restaurant serves lobster, and that John left a big tip, the reader can also guess that the restaurant was probably of the fancy type rather than a fast-food type, and that the food was probably good. If asked to elaborate, the reader might come up with the following expanded paraphrase:

`John went to MaMaison. The waiter seated John. The waiter gave John the menu. John asked the waiter for lobster. John waited for a while. The waiter brought John the lobster.`

1. The terms "distributed neural networks," "parallel distributed processing (PDP)," and "subsymbolic" are used interchangeably in this book to refer to neural network models that process distributed representations of data. The term "connectionist" is used in a wider sense that also includes models based on local representations (distributed and local representations are discussed in section 2.2).

```
John ate the lobster. The lobster tasted good. John paid
the waiter. John left a big tip. John left MaMaison.
```

Or the reader could answer questions about the stories in the following way:

```
Q: What did John buy at Radio-Shack?
A: John bought a CD-player at Radio-Shack.
Q: Where did John take a plane to?
A: John took a plane to JFK.
Q: How did John like the lobster at MaMaison?
A: John thought the lobster was good at MaMaison.
```

The above examples illustrate a number of issues that make script-based story understanding an interesting task. The answers and the paraphrase show that the reader makes a number of inferences beyond the original story. The inferences are not based on specific rules but on statistical regularities, learned from experience. The reader has experienced a number of similar event sequences in the past and the unmentioned events have occurred in most cases. They are assumed immediately and automatically, and quickly become part of the memory about the narrative. If the reader is asked later whether John actually flew to JFK, he would quite confidently confirm this, but might not be able to tell whether this event was actually mentioned in the story or only inferred (Bower et al. 1979).

Another group of issues concerns episodic memory. Narratives are stored in memory one at a time as they are read in, with only a single presentation. We do not usually have to go back and reactivate previous narratives after we have read a new story. However, the new story is recognized as an instance of a familiar sequence of events, and attention is paid only to the facts that are specific to this story. It seems that episodic memory is structured to support classification based on similarities and storing the differences, and that this structure has been extracted from experience (Kolodner 1984).

Episodic memory structure also supports associative retrieval. A question supplies only partial information about the story it refers to, yet the story is retrieved with only the question as a cue. The more unique the story is in the memory, the less needs to be specified in the question. If there is only one travel story, we can unambiguously ask `Where did John travel to?` Usually, if there are several alternative stories, the most recent one is recalled by default. Context of the previous question can also be used to select among the alternatives (Lehnert 1978). And of course, it is possible to recognize a situation where there is nothing appropriate in the memory.

Issues concerning the lexical and semantic memory are perhaps less obvious but very fundamental. A cognitive model of natural language understanding must be able to represent word meanings effectively, because they constitute the basic currency of the model. The meanings could be coded in by hand, using some elaborate representation scheme. However, it would be preferable to *learn* the meanings from examples. The properties of the words most crucial to the task should be extracted and coded into the representation automatically. Finally, a mechanism for mapping the lexical word symbols to these meanings is necessary. This is a many-to-many mapping: some lexical words correspond to several concepts, and the same concept can usually be expressed with several different words.

DISCERN is a distributed neural network model specifically designed to deal with the above issues. More generally, DISCERN constitutes an integrated approach to subsymbolic natural language processing. Separate distributed neural network models of various subtasks are brought together into a single high-level system that is similar in scope to the traditional symbolic models.

1.2 *Motivation and Goals*

Understanding stories about stereotypical event sequences is not a new task. Script theory was developed to explain how knowledge about familiar everyday routines is used in story understanding (Schank and Abelson 1977). At the same time, symbolic computer models were built that implemented the theory at various levels (Cullingford 1978; DeJong 1979). These models exhibit quite impressive behavior in their domains and they are certainly capable of processing simple script instantiations such as the above examples.

What makes this task worth doing with artificial neural networks? There are several issues that the symbolic approach does not address. The processing architecture, mechanisms, and knowledge in symbolic systems are hand-coded with a particular domain and data in mind. Inferences are based on handcrafted rules and representations of the scripts. Such systems cannot utilize the statistical properties of the data to enhance processing. They can only do what their creator explicitly programmed them to do. This contradicts the intuitive notion of scripts. They should not be preset, fixed, all-or-none representations, but instead should emerge automatically from the statistical regularities in the experience.

The symbolic models are high-level process models, far removed from the physical structures that implement the processes in the brain.

As a result, they inherently lack the capability of explaining certain aspects of human performance. In the symbolic framework, it is very difficult to address issues such as: Where do performance errors come from? How can memory become overloaded and why do certain types of memory confusions occur in overload situations? What happens when the system is corrupted with noise, or when parts of it are destroyed?

A major motivation for DISCERN is to give a better account of such high-level cognitive phenomena in terms of the special properties of distributed neural networks such as learning from examples, automatic generalization, and graceful degradation. Specifically, DISCERN aims at showing how script-based inferences can be learned from experience, based on the statistical correlations in the example data; how the episodic memory organization can be automatically formed based on the regularities in the example data; and how word semantics can be learned from examples of their use. The model should give plausible explanations of how expectations and defaults automatically emerge. Because artificial neural network models are motivated by the physical structures in the brain, they have a chance of better explaining performance errors and deficits. DISCERN should exhibit plausible memory interference errors, role-binding errors, and lexical errors in overload situations and in the face of noise and damage.

On the other hand, DISCERN is a demonstration that a large-scale natural language processing (NLP) system that performs at the level of symbolic NLP models can be built from distributed artificial neural networks. A script-processing system needs to read its input stories word by word in natural language. It needs to process meaningful internal representations where all inferences are explicit, so that its behavior is interpretable to the external observer. It needs to have long-term memories both for the general, statistical knowledge about scripts and word semantics and for the specific, exact knowledge about actual stories with actual word tokens. The model should be able to produce fully expanded paraphrases of stories, including all inferences. It should be able to answer questions about specific events of a specific story in the memory. Finally, the output should be produced word by word in natural language.

Script understanding forms an excellent testbed for research in neural network techniques. Much of the research in network architectures and algorithms has been "puzzle solving" without clear high-level motivation, and often the results are not relevant to the central issues in cognitive modeling. I believe that by using a concrete high-level problem to identify technical issues and to motivate solutions to them, we

are more likely to develop techniques that lead to true progress. This approach also makes us well aware of the limitations of these techniques.

Therefore, the third major goal of DISCERN is to develop general distributed neural network mechanisms for dealing with the complexity and structure of high-level cognitive modeling. These include modular network architectures, automatic learning of distributed representations, methods for one-shot learning, hierarchical self-organization, many-to-many mapping, role binding, and type/token processing. These techniques are designed to solve problems in script processing, but should prove useful in other high-level tasks as well.

1.3 Approach

Previous research in parallel distributed processing (PDP; see e.g. McClelland et al. 1986b; Rumelhart et al. 1986c for an overview) has concentrated mostly on isolated, small, low-level tasks, and relied heavily on pre- and postprocessed data. In many cases, the problem is reduced to learning a simple mapping. As a result, such PDP models typically have very little internal structure. They produce the statistically most likely answer given the input conditions, in a process that is opaque to the external observer. This suits modeling isolated low-level tasks, such as learning past tense forms of verbs (Rumelhart and McClelland 1986) or pronunciation of words (Sejnowski and Rosenberg 1987). However, modeling higher-level cognitive tasks with homogeneous networks has been infeasible, for three reasons:

1. High-level tasks are often composites of distinct subtasks. They consist of several interacting subprocesses, such as parsing language, generating language, memory storage, memory retrieval, and reasoning. Complex behavior requires bringing together several different kinds of knowledge sources and processes, which cannot be done in a single pattern transformation. Such behavior requires structured architectures (Feldman 1989; Minsky 1985; Simon 1969).
2. The required network size, the number of training examples, and the training time become intractable as the size of the problem grows (Cottrell and Tsung 1989; Elman 1991b; Harris and Elman 1989; Jain 1991a; Servan-Schreiber et al. 1989; St. John and McClelland 1990).
3. There is no way to evaluate what the entire system is doing, for example, what knowledge it is acquiring and applying, unless

each module processes meaningful internal representations that can be interpreted by an external observer and by other modules in the system.

A plausible approach for high-level cognitive modeling, therefore, is to construct the architecture from several interacting modules that work together to produce the high-level behavior (for related arguments for modularity, see Fodor 1983; Minsky 1985; Shallice 1988). Central issues to be addressed in this approach are:

1. How the overall task should be broken into modules and how the modules should be organized;
2. What are the appropriate architectures for different kinds of subtasks;
3. How PDP subnetworks should be designed so that they can serve as modular building blocks;
4. How communication between such building blocks can be established; and
5. How to make use of the modular structure in training the system.

A complex architecture will need to have a common set of terms, serving the function of a "symbol table" for intercommunication. In a large system consisting of many modules and many communicating pairs, this function is most efficiently established through a global vocabulary, a central lexicon. The modules communicate using terms from this global symbol table, instead of having a separate set of terms for each communication channel. Each module can then be trained separately and in parallel, as long as they are trained with compatible input/output (I/O) data.

In a high-level subsymbolic model, communication (i.e. input and output of each module) can take place using distributed representations. In backpropagation networks with hidden layers, the network automatically develops internal distributed representations for the I/O items as a side effect of learning the processing task (Elman 1990; Hinton 1986; Miikkulainen and Dyer 1987; Rumelhart et al. 1986b). These representations reflect the regularities of the task and data, extracted without external supervision, and provide an excellent basis for generalization. Central to this book is a mechanism for forming distributed representations, called FGREP (Miikkulainen and Dyer 1987, 1988, 1989a, 1991), where the internal representations discovered by the network are made public in a global lexicon, so that they can be used for communication in a large modular system.