XIII. COGNITIVE INFORMATION PROCESSING

Academic and Research Staff

Prof. Murray Eden
Prof. Jonathan Allen
Prof. Barry A. Blesser
Prof. Francis F. Lee
Prof. Samuel J. Mason
Prof. William F. Schreiber
Prof. Donald E. Troxel
Prof. Ian T. Young

Dr. Jack E. Bowie
Dr. Ralph W. Gershberg
Dr. Uri F. Gronemann
Dr. Martin J. Hewlett*
Dr. Conrade C. Jaffe†
Dr. Perry L. Miller
Dr. Mary J. Naus‡
Dr. David M. Ozonoff

Dr. Stephanie E. Sher
Francis X. Carroll
Jeffrey A. Hagerich
M. Sharon Hunnicutt
Eric R. Jensen
Kenneth A. Okin
Robert A. Piankian
Kenneth P. Wacks

Graduate Students

Brian E. Boyle
Becky J. Clark
Michael A. Cohen
Charles H. Cox
Caleb J. Drake
James R. Ellis, Jr.
Irvin S. Englander
Ali M. Fakhr
Steven G. Finn
Harry S. Forsdick
Alan M. Gilkes
Richard S. Goldhor
David B. Harris

Marvin E. Jernigan
Clayton K. S. Kau
Richard C. Kinnaird, Jr.
John W. Klovstad
Theodore T. Kuklinski
Sze-Ping Kuo
Donald S. Levinstone
Charles W. Lynn
Lee C. Makowski
Dennis V. Marsicano
Douglas O'Shaughnessy
John E. Ostrowski
Douglas B. Paul
Roger S. Putnam

Robert J. Shillman
Robert D. Solomon
Jan A. Spriet
Christopher E. Strangio
Hoo-min D. Toong
Margaret A. Turek
Bik-Kwoon Tye
Joseph E. Wall, Jr.
Allen W. Wiegner
Harvey M. Wolfson
Kenway J. Wong
Tonny Y. Wong
Gregory W. Zack

A. TEXT-TO-SPEECH CONVERSION

1. LOCALLY ORGANIZED PARSER FOR SPOKEN INPUT

Madeline Moses Fund

Perry L. Miller

A Locally Organized PARSer for Spoken Input, (LPARS) has been implemented on the TX-2 computer at the M. I. T. Lincoln Laboratory. It is designed to process continuous speech with the help of syntactic and semantic information.

The LPARS system differs from traditional parsing methods in that it has no inherent left-to-right, or right-to-left, bias to its operation. Rather, it allows syntactic structures to be recognized locally in any part of an utterance. In fact, several parse structures may be built up simultaneously in different parts of the sentence, and later connected by searching for words that might reasonably exist between them.

Thus words and phrases reliably recognized in any part of a sentence can be used

*Research Affiliate from the Department of Biology.

†Picker Foundation Fellow.

‡Also Assistant Professor of Psychology at Wellesley College.

to help guide the search for further words to complete the sentence.

Task Domain

The vocabulary of LPARS contains approximately 70 words. It recognizes a very restricted, but linguistically natural and interesting, subset of English. Its semantics are defined in terms of a particular scene – a small two-room house containing people, furniture, fixtures, etc. – about which one may make statements, ask questions, tell a very simple story, or command the system to manipulate the scene. Sample input sentences are:

"The coffeetable on which the ashtray is placed by Robert supports the dictionary."

"What does the sidetable support?"

Operation

LPARS expects as input a string of phoneme candidates from a front-end phoneme recognizer. For the work reported here the input was prepared by a phonetic scrambler program that simulated front-end behavior, rather than by a real phoneme recognizer. The input phoneme candidates may be ambiguous (i.e., several possibilities may be given for one segment). The input is also expected to contain a fairly large amount of error.

LPARS operates by first making an initial scan through a sentence looking for longer words that are not too garbled. The scan returns a list of word candidates that match within a given error tolerance to specified sections of the input. It is expected that some of these word candidates are incorrect, and indeed some may overlap.

These word candidates are turned over to the higher level part of the system which initiates scans for small words and for more highly garbled words in the areas adjacent to and between the words found in the initial scan, and groups the words together into parse structures. This processing is done systematically in an attempt to uncover the entire utterance.

Simplified Example

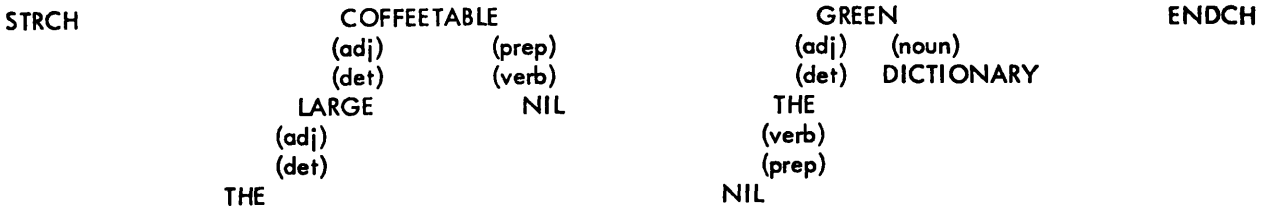
Figure XIII-1 is a simplified example of LPARS in operation. The input utterance is the sentence: "The large coffeetable supports the green dictionary." In real operation, the sentence would be spoken, analyzed by a front-end phoneme recognizer, and then a string of phoneme candidates would be produced for input to LPARS. At present, this input is produced by a front-end simulator.

In an initial scan through the sentence, LPARS tests the words in its vocabulary against the entire length of the input. This scan is made at a low phonetic distance looking for longer words that are not too garbled. (Phonetic distance is a measure of

INPUT SENTENCE:
'THE LARGE COFFEETABLE SUPPORTS THE GREEN DICTIONARY'

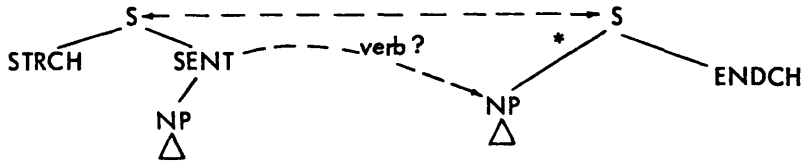
(1) INITIAL SCAN: COFFEETABLE, GREEN

(2) LOCAL HIGHER DISTANCE SCANS:



(3) PARTIAL PARSE TREE CONSTRUCTION:

(4) CONNECTION OF PARTIAL PARSE TREES:



(5) RECOGNIZED SENTENCE:

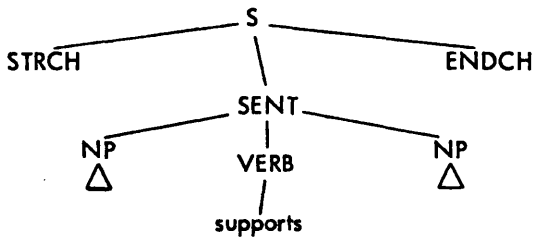


Fig. XIII-1. Simple example of LPARS in operation.

how closely the phonetic spelling of a word matches a section of input.) Let us assume, in this instance, that the two word candidates "coffeetable" and "green" are found. These word candidates are turned over to the high-level part of the system, together with a start-of-sentence character (STRCH) and an end-of-sentence character (ENDCH) which is added by the system.

The higher level analysis (Fig. XIII-1) includes three fairly distinct stages. First,

scans at higher phonetic distance are made based on fairly local cues, in the areas adjacent to the word candidates. In this example, these local scans are very simple. A scan in front of the noun uncovers the adjective "large" and a further scan in front of that word uncovers the determiner "the." A scan to the right of the noun for prepositions and verbs fails to find any word candidates. This means that the verb "supports" is too garbled to be picked up by either the initial scan or the higher phonetic distance selective scan. Similar scans around the adjective "green" uncover the words "dictionary" and "the," but again fail to find the verb.

After these higher distance scans have taken place, the system builds up as many parse trees as it can in the sentence. If all words in the sentence have been found, then the entire sentence is constructed. Otherwise, the result is a number of partial parse trees (PPTs) in different parts of the sentence.

The system attempts to construct as many such PPTs as it can with the words it has found. In this example, it constructs only two PPTs. The first tree, "STRCH the large coffeetable," is straightforward; the second, "the green dictionary ENDCH," is somewhat unusual, since it contains an "ancestor link" (the link labeled "*"). The ancestor link allows LPARS to recognize that, because of right recursion, many syntactic relationships are possible, but to defer commitment until later, when an attempt is made to join this structure to another structure by proposing words between them.

The third and final step in the parsing process is to connect PPTs to one another by using the grammar to propose words that might exist between them. Any words proposed are tested against the input at even higher phonetic distances than the previous scans.

In the present example, the algorithm discovers that the two parse trees can be connected if a verb is found between them. It therefore initiates a higher distance scan which succeeds in finding the verb "supports." Thus the entire sentence is recognized.

Notice that this example is a simplified one. No erroneous words were found. In a more realistic example, erroneous words would be found, some local structures containing these words would be built up, and, additionally, some erroneous local parsings of the correct words could be constructed. Attempts to build out upon the erroneous structures usually prove unsuccessful, while usually attempts to build out upon the correct structures eventually succeed.

Experimental Evaluation

The LPARS system was evaluated by processing 50 sentences. These input sentences were produced by a front-end simulator which approximated the accuracy of the Vicens-Reddy phoneme recognizer. The simulator operates roughly as follows: 15% of the phonemes are deleted; 10% of the remaining phonemes are badly scrambled (i.e., a stop might be changed into a vowel or fricative); for the remaining phonemes substitutions

are made at random from a restricted class of phonemes (i.e., a stop might be changed into some other stop chosen at random).

During the evaluation, the recognition of the sentence was considered successful if the correct sentence was included among several possible sentences that were found. When several sentences were found, the correct sentence was almost always the best match.

During evaluation of LPARS, the 50 sentences were processed with the following results: Twenty-three sentences were correctly recognized from words found only by the initial local processing; nineteen sentences were correctly recognized by the PPT connection algorithm, after the initial local processing; and three sentences were correctly recognized by fallback methods. Two sentences resulted in incorrect recognition. In both cases, the sentence found differed from the input sentence in a single content word. Finally, three sentences resulted in failure to find any possible sentence at all. Thus the overall success rate of LPARS with the 50 input sentences was 90%. We conclude that, given a simulation of a fairly crude front end, the ideas embodied in LPARS can be made to work acceptably.

This system was developed by the author and the research submitted to the Department of Electrical Engineering, M.I.T., in March 1973, in partial fulfillment of the requirements for the degree of Doctor of Philosophy. It has been published as Lincoln Laboratory Report 503.