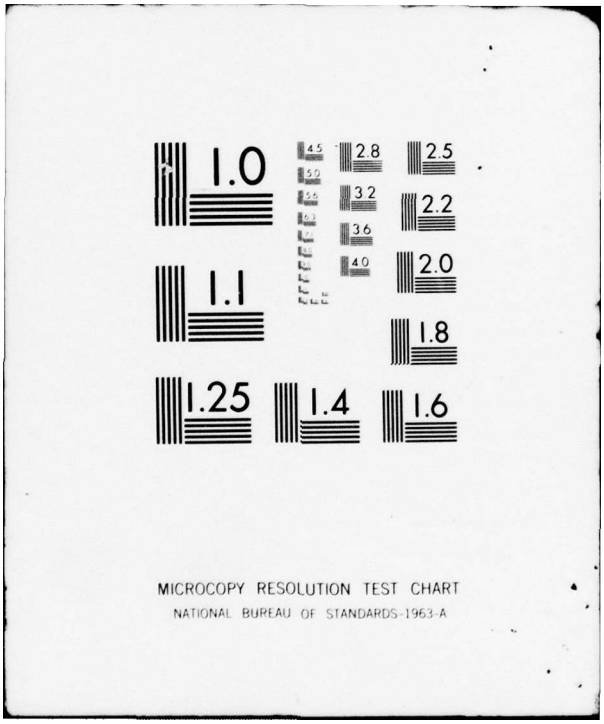
AD-A056 231 LOGICON INC SAN DIEGO CALIF F/G 9/2
LISTEN: A SYSTEM FOR RECOGNIZING CONNECTED SPEECH OVER SMALL, F--ETC(U)
APR 78 J E PORTER N61339-77-C-0096
UNCLASSIFIED NAVTRAEQUIPC-77-C-0096-1 NL

1 OF 2
AD A056231




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A056231
DDC FILE COPY
AD No.

LEVEL II

12

Technical Report NAVTRAEQUIPCEN 77-C-0096-1

LISTEN:
A SYSTEM FOR RECOGNIZING CONNECTED SPEECH OVER SMALL, FIXED VOCABULARIES, IN REAL TIME

Logicon, Inc.
San Diego, California 92138

FINAL REPORT JULY 1977 - MARCH 1978

April 1978

DoD DISTRIBUTION STATEMENT

Approved for public release;
distribution unlimited.

DDC
RECEIVED
JUL 14 1978
E

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

78 07 12 077

NAVTRAEQUIPCEN 77-C-0096-1

GOVERNMENT RIGHTS IN DATA STATEMENT

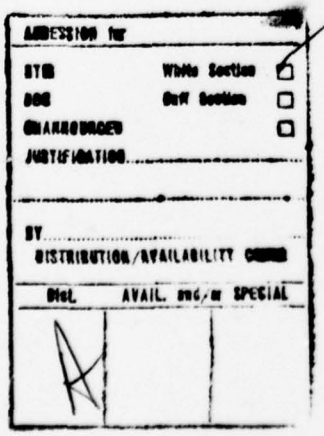
Reproduction of this publication in whole or in part is permitted for any purpose of the United States Government.

FOREWORD

The development of a limited continuous speech recognition (LCSR) system is seen as a necessary step for the widespread use of automated speech technology. Commercial isolated word recognition (IWR) systems have certainly proved to be highly reliable and accurate. However, certain tactical situations require speed as well as accuracy with short system pretraining and familiarization. Further, operator performance is the primary goal of the system. This means that the system must be capable of accurately recognizing the entire vocabulary because what is said can be a cue to how the operator is learning the task. Thus, a system is required for training which is accurate, which can handle continuous speech with very little pretraining, and has a branching factor equal to the size of the vocabulary.

This report concerns one approach toward that goal. Preliminary results indicate potential for commercial IWR systems to be used in LCSR.

ROBERT BREAUX
Scientific Officer



78 07 12 077

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NAVTRAEQUIPC 77-C-0096-1	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): LISTEN: A System for Recognizing Connected Speech Over Small, Fixed Vocabularies, In Real Time	5. TYPE OF REPORT & PERIOD COVERED: Final Report, July 1977 - March 1978
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): J. E. Porter	8. CONTRACT OR GRANT NUMBER(s): N61339-77-C-0096
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Logicon, Inc., P.O. Box 80158, San Diego, California 92138	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 3753-5P2
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Training Equipment Center, Code N-71, Orlando, Florida 32813	12. REPORT DATE: April 1978
	13. NUMBER OF PAGES: 100 (98 p.)
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Speech Recognition
Speech Understanding
Continuous Speech Recognition
Controller Training System
Mathematical Machines
Real-time Speech Recognition
Connected Speech Recognition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the development of a system for recognizing connected speech in real time using a commercially available speech preprocessor, a minicomputer and programs written in FORTRAN. The system was tested on two speakers using the digits and the word "point" with inconclusive results. Recognition accuracy of 86% percent was achieved for one speaker whereas accuracy for the other speaker was lower (39% percent) due to an anomalous difference between training and test data for that speaker's voice.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

406 012

TABLE OF CONTENTS

Section		Page
I	INTRODUCTION	5
	Background	5
	Historical Review of the LCSR Project	7
	Overview of LCSR Reports	9
II	SUMMARY OF TASKS PERFORMED	12
	Summary of Phase 0	12
	Analysis of Requirements	12
	Recognition Technique and Available System Survey	12
	Description of the Mathematical Machine Approach to LCSR	13
	A Phased LCSR Project Plan	14
	Feature Selection	14
	Voice Data	14
	Example Space Generation	15
	Structure Extraction	15
	Tests on Structural Information	15
	Summary of Phase 1	16
	Rigorous Test of Structural Violation	16
	Analysis of Structural Violations	16
	Recognition Algorithms which Accommodate Violations	16
	Dynamic Programming Formulation	17
	Sound Duration Quality Measures	17
	Real-Time Word Spotter	18
	Word Interaction Model	18
	Word Selection Algorithm	18
	Statistical Models	18
	Real-Time Word Selection	19
	LISTEN Tests	19
	Additional Speakers	19
	Analysis of Errors	19
III	MEX, THE WORD-SPOTTING PORTION OF LISTEN	20
	Development of the Algorithm	20
	Accommodating Loop Letter-Set Violations	22
	Accommodating Transition Letter-Set Violations	23
	Controlling Proliferation of Copies	27

TABLE OF CONTENTS

Section		Page
	General Description of Dynamic Programming Machine Operation	29
	Copy Initialization	31
	Final Stage Processing	32
	Dropping Criteria	34
	Quality Functions	35
	Description of Algorithm	39
	Input	39
	Output	39
	Processing	39
IV	MINT, THE WORD SELECTING PORTION OF LISTEN	53
	The Word Interaction Problem	53
	Observables	54
	Key Relationships	55
	Formulation of the Word Interaction Problem as One In Statistical Decision Theory	57
	An Important Transformation	69
	Solution of the Problem by Dynamic Programming	71
	Real-Time Considerations	73
	Description of the MINT Algorithm	74
	Flow Charts	75
	Computing Cost Contributions	80
	ΔQ^{ap}	80
	ΔQ^{I}	81
	ΔQ^{A}	83
	ΔQ^{G} (Start, π)	85
	ΔQ^{G} (m, m')	86
	ΔQ^{G} (m, End)	88
V	RESULTS, CONCLUSIONS AND RECOMMENDATIONS	89
	Results	89
	Real-Time Operation	89
	Recognition Accuracy	90
	Analysis of Errors	90
	Conclusions	94
	Recommendations	94

LIST OF ILLUSTRATIONS

Figure	Title	Page
1	The Non-Deterministic Finite Transducer Developed in Phase 0.	20
2	Modification of the Non-Deterministic Finite Transducer to Accommodate L-Set Violations	23
3	Modification of the Non-Deterministic Finite Transducer to Accommodate Transition Letter-Set Violations	26
4	Unmodified and Modified Last Stage Processing.	34
5	Outer Loop	41
6	Process Machine Type	42
7	Process Copies in Last Stage	43
8	Process Copies in Typical Stage	44
9	Advance a Single Copy.	45
10	Advance One of Several Copies	46
11	Process a Single Copy which Remains in Stage without Offspring	47
12	Process Several Copies which Remain in Stage, without Offspring	48
13	Process a Single Copy with Potential for Producing an Offspring	50
14	Process Several Copies with Potential for Producing Offspring	51
15	Potential Start Processing	52
16	Directed Graph Representing Real Recognitions and Artifacts, and the Potential Immediate Successors Relation	59
17	Annotated Directed Graph Depicting the Data Comprising an Observation of an Utterance	63
18	Receive a New Node	76
19	Move This Node from Group III into Group II	77
20	Move the Oldest Node of Group II out of Group II	78
21	Do End of Utterance Processing	79
22	The Distribution Assumed for Q_L	82

LIST OF ILLUSTRATIONS

Figure	Title	Page
23	The Distribution Assumed for Inter-Word Gaps	87
24	Confusion Matrix — 1049 Words (MWG Test Data) 49 Word Mistakes, Including Insertions, Deletions, and Substitutions 95.3% Correct	91

LIST OF TABLES

Table	Title	Page
1	Quality Category	37

SECTION I

INTRODUCTION

This report documents the results obtained and work accomplished during the second phase (Phase 1) of a Limited Continuous Speech Recognition (LCSR) project. The ultimate goal of this project is to obtain, at the earliest possible opportunity, a capability that meets the requirements of the Naval Training Equipment Center (NAVTRAEQUIPCEN), for application in training systems.

Background

LCSR is generally understood in the speech research community to mean the problem of automatically recognizing natural human speech consisting of isolated utterances which are sequences of words chosen from a small (less than 30-word) vocabulary spoken continuously; i.e., without pauses or breaks between words. Reliable automatic recognition of isolated words is now a fait accompli, but the techniques used in isolated word recognition have proven to be ineffective when applied to continuous speech. Since continuous speech is the more natural mode of human verbal communication, the scientific and industrial communities are vigorously pursuing extensions of earlier successes in isolated word recognition. However, the general problem of automatically recognizing continuous speech has shown itself to be so difficult as to warrant restriction to a limited problem, focusing on the difficulties involved in reliably recognizing even relatively few words in relatively short utterances.

NAVTRAEQUIPCEN has demonstrated the gains to be made in training effectiveness by incorporating an automatic speech recognition capability in training systems. Isolated word recognition is often sufficient to meet the

requirements of these systems. But it also often occurs that an automated training system either requires, or can be made much more effective by, introducing an LCSR capability. NAVTRAEQUIPCEN therefore supported this project to investigate the possibilities for obtaining an LCSR capability in support of training system development.

The initial goals of the LCSR project were:

a. To identify the unique features of the LCSR problem as it arises in training applications, in order to have the sharpest possible definition of the problem of interest to NAVTRAEQUIPCEN, and to guide the remainder of the project.

b. To survey the relevant technical literature and industry to determine the state of the LCSR art, and to determine if there exist systems or techniques which satisfy the requirements of the training LCSR problem.

c. On the basis of an understanding of the training LCSR problem and the current state of the LCSR art, to assess the feasibility of obtaining an LCSR capability for support of training system development.

d. If obtaining a training LCSR capability is feasible, to develop a plan for doing so.

Early work in pursuit of these goals led to the important conclusion that development of a training LCSR capability, while not without considerable difficulty, was a worthwhile venture. This conclusion was reached in part because a new approach to LCSR, called the mathematical machine approach, was formulated. This approach was subjected to preliminary tests, with promising results, during the initial phase (Phase 0) of the LCSR project. For a detailed description of the LCSR problem as it arises in training applications, a review of the state of connected speech recognition at the beginning of the project, and an introduction to the technical details of the mathematical machine approach to LCSR, see the Phase 0 Final report; Use

of Computer Speech Understanding in Training: A Preliminary Investigation of a Limited Continuous Speech Recognition Capability. NAVTRAEQUIPCEN Report 74-C-0048, June 1977.

Historical Review of the LCSR Project

The survey of technical literature and industrial accomplishments in connected speech recognition, prepared early in Phase 0 of this project, revealed that no technique for performing LCSR suitable for application to training systems of interest to NAVTRAEQUIPCEN had been reported or was available off-the-shelf. The primary shortcomings of reported techniques and systems available at that time were severe. Various approaches:

a. Were incapable of operating in real time on minicomputers

b. Required very expensive, specialized equipment (preprocessors or parallel processors),

c. Used higher order information sources, such as semantic and syntactic constraints which are not necessarily present in training applications,

d. Did not achieve adequate recognition accuracy, or

e. Most frequently, were some combination of the above.

On the other hand, progress towards viable continuous speech recognition was clearly discernible in an historical review of the literature. Various techniques had been described which, individually, could meet each of the unique requirements for LCSR in the training environment. Awareness of this state of affairs led to the formulation, at Logicon, of the mathematical machine approach to LCSR.

This approach adopts a sequential decoding technique for word-spotting as its first component in order to make real-time operation at least potentially feasible. Fast operation requires that the sequential model of the words to be recognized must be simple, unlike those used at Carnegie-Mellon University

or at IBM. In this sense, the approach to word spotting has more in common with the techniques described by Martin in his dissertation of 1970[1]. However, the adopted approach relies heavily on automatic extraction of the simple sequential characterization of vocabulary items to be recognized from examples in a training sample, in contrast to Martin's painstaking manual derivation of the word characterizations. There are other profound differences in the effective feature space used, the use of temporal characteristics and, most importantly, treatment of interactions among potential recognitions. In regard to the last mentioned difference (interaction) the adopted approach resembles the HARPY system developed by Reddy and Lowerre[2], in that both solve a decision theoretic problem on a directed graph describing the utterance, using methods of dynamic programming.

The mathematical machine approach to LCSR-formulated by Logicon is uniquely matched to expected training system requirements in several ways. In particular, this approach is specifically suited to real-time operation. In addition, (unlike HARPY), it avoids consideration of higher knowledge sources, as in some applications (such as recognizing strings of digits) these sources are essentially absent. The adopted approach also accepts a significant off-line computational burden for characterizing individual speaker's voices, a feature which is unacceptable in many applications but often benign in the training environment. The adopted approach is also suitable for use with the speech preprocessors available at the NAVTRAEQUIPCEN, with which considerable experience has been accumulated.

---

1. T. B. Martin (Univ. of Penn.), "Acoustic Recognition of a Limited Vocabulary in Continuous Speech," Dissertation, Department of Electrical Engineering, Univ. of Penn., 1970.

2. B. T. Lowerre (CMU), "The HARPY Speech Recognition System," Dissertation, Department of Computer Science, CMU, April 1976.

An initial investigation of the mathematical machine approach to LCSR was thus undertaken because it was uniquely matched to the speech recognition requirements inherent in training applications. This initial investigation sought to validate the assumptions upon which the approach was predicated, most notably that the output of a deterministic preprocessor carries enough information to support reliable recognition of continuous speech. The output information must be present in such a way that individual words are represented:

a. By characteristic classes of output,

b. In a fixed order,

c. With characteristic time durations and time intervals between characteristic output samples.

Also, the characterizing classes of output (and their order) must successfully be extracted automatically from training data. The preliminary investigations were carried out in Phase 0 of the LCSR project, and very promising results were obtained. Briefly stated, it was shown that the preprocessor output does have a rich sequential information structure, that it could be extracted automatically, that the extracted structure occurred reliably, and that it had utility for recognition.

Phase 1 of the LCSR project was thus instituted based on the promising indications of Phase 0 results. The primary goal of Phase 1 was to complete the development of a preliminary LCSR system, using the mathematical machine approach and to determine if the underlying assumptions were valid for additional speakers. The primary accomplishment of Phase 1 was thus the production of LISTEN, Logicon's Initial System for the Timely Extraction of Numbers; a minicomputer based system for recognizing continuous speech in real time.

Overview of LCSR Reports

The Phase 0 final report and this, the Phase 1 final report, together describe the development of Logicon's approach to LCSR, from the analysis of special requirements unique to the training environment and formulation of the approach, through subsequent development and initial implementation in LISTEN.

Many concepts and terms used in this report are introduced and defined in the Phase 0 final report. The discussions of the recognition algorithm given in this report, in particular, presuppose familiarity with concepts such as transition and loop letter sets and the finite automaton characterization of a sequential recognition procedure. Readers interested in the technical details of LISTEN must therefore read both final reports.

The Phase 0 final report contains:

a. A discussion of requirements for recognition of continuous speech in support of training,

b. A survey of the technical literature pertaining to connected speech recognition,

c. A description and rationalization of Logicon's approach to LCSR,

d. A description of the speech data, used in the project data gathering procedures, and extraction of individual examples of vocabulary items from continuous speech examples,

e. A description (with examples) of the sequential information structure extracted from one speaker's speech data.

f. A description of the results of several preliminary tests of the extracted information structure for automatic recognition.

This report contains:

a. A brief review of the technical work accomplished during Phase 0, and a more comprehensive review of the technical work accomplished in Phase 1 (Section II).

b. A description of the development of MEX, the word-spotting portion of LISTEN, including citation of the analyses and findings which determined the salient features of this portion of the algorithm, (Section III),

c. A description of the development of MINT, that portion of the recognition algorithm which treats the relationship among potential recognitions detected by MEX. A mathematical formulation of the interaction problem is given which motivates the description of MINT as a method of solving a problem in statistical decision theory via dynamic programming (Section IV),

d. Results of preliminary tests of LISTEN, conclusions drawn from those test results, and recommendations for completing the investigation of this approach to LCSR.

SECTION II

SUMMARY OF TASKS PERFORMED

Summary of Phase 0

Analysis of Requirements — The source of the requirement for a continuous speech recognition capability in support of automated training was reviewed. The training environment was found to frequently impose a unique set of requirements on a supporting LCSR capability; specifically

a. Real-time, or very near real-time operation is necessary.

b. Small vocabularies are often adequate.

c. The vocabulary is often fixed, or changes infrequently.

d. Recognition accuracy must be high.

e. Semantic, syntactic and task-related higher knowledge sources are sometimes completely missing, as in the string of digits case.

f. Speaker independence, while convenient, is not necessary.

g. System cost must be fairly small; for example expensive specialized preprocessors should be avoided, and the computational burden should be compatible with minicomputer technology.

Recognition Technique and Available System Survey — The technical literature was searched for reports of recognition techniques that would satisfy the unique requirements of LCSR for training. Existing commercially available speech systems were also surveyed to determine if any met these requirements. It was found that neither recognition techniques nor complete systems which could clearly meet the requirements were available. However, interesting progress was evident in automatic speech recognition

research and several systems were found that partially met the requirements. Most notable among these were the HARPY system, developed by Reddy and Lowerre at Carnegie-Mellon which had impressively high recognition accuracy (but operated at many times real time on large computers) and a system described by Martin in his dissertation of 1970, which operated in real time (but was implemented in hardware, based on a painstaking manual analysis of samples of the vocabulary items to be recognized).

This survey of reported techniques left a clear impression that a proper combination of existing techniques might meet the special requirements of LCSR for training applications. It also showed that many approaches to connected speech recognition were being followed, each suited to a different set of application requirements, and that direct progress towards an LCSR capability geared specifically to the unique LCSR for training requirements, would probably only occur in a program consciously aimed at those unique requirements.

Description of the Mathematical Machine Approach to LCSR – A technique for LCSR, formulated by Logicon, was described. This technique, called the mathematical machine approach, is based on the assumption that any speech preprocessor which can support LCSR must encode spoken vocabulary items in its output:

a. in reliably occurring classes of output

b. in a fixed order

c. with characteristic time duration.

Under this assumption, a sequential decoding recognition procedure for spotting potential occurrences of the vocabulary items is used, and discrimination of real and artifactual recognitions is done subsequently, at a reduced data rate. The mathematical machine approach entails this two-part

algorithm and the automatic extraction by a specialized, partially heuristic algorithm, of the reliably occurring information structure on which the sequential word spotting algorithm is based.

The mathematical machine approach was shown to match the unique requirements of LCSR for training, and thus be worthy of initial investigation.

A Phased LCSR Project Plan – A phased development plan was created for investigation of the mathematical machine approach to LCSR. The completion of Phase 0 was devoted to validation of the assumptions on which the approach is based, with respect to the speech preprocessor in use at NAVTRAEQUIPCEN (The Threshold Technology Model VIP-100). If those assumptions were found to be valid, the plan called for further investigation of the feasibility of developing an LCSR capability based on the mathematical machine approach.

Feature Selection – Fifteen of the thirty-one binary features derived by the speech preprocessor were chosen for use in the LCSR project. These are the features that indicate the presence in the input voice signal of various phoneme categories or phoneme groups, such as nasals or unvoiced noise-like consonants. One spectral feature (number 17) is included, which indicates energy concentration in a passband centered near 5 KHz.

The features were chosen on the basis of visual analysis of many samples of VIP-100 output. It was found that regular patterns of presence and absence of these features could be detected by eye. The features not selected are indicators of energy concentrations in spectral bands, and were less reliable indicators of the presence and absence of vocal gestures.

Voice Data – A large body of continous speech data were collected for a single speaker (MWG). The utterances consisted of one to four words, balanced across the vocabulary (the digits and the word point). The data were

collected over a two-month period and divided into three equal groups called Training, Interim Test and Test data. Test data were kept only on disc and not accessed in any way during the development of the recognition algorithm.

Example Space Generation – Both automatic and manual methods were used to extract from the recorded multi-word utterances, VIP-100 output examples containing single words with a minimum of extraneous material. The collections of examples of individual vocabulary items thus formed were called example spaces.

Structure Extraction – The structure extraction algorithm mentioned earlier was coded and applied to MWG's example spaces. Reliably occurring sound groups were found for each vocabulary item, showing both that the extraction algorithm worked well, and that the VIP-100 output, for MWG's voice at least, carried a large amount of reliably occurring structural information. Another algorithm was designed, coded and applied to the speech data to characterize the sound groups occurring between the highly reliable points within words. The structure thus found is represented by transition and loop letter sets.

Tests on Structural Information – The transition and loop letter sets were tested in several preliminary ways as the last activity of Phase 0. The transition letter sets were tested to determine the reliability with which they occurred in non-training data, and their ability to distinguish real occurrences of the vocabulary items. Although preliminary, all of these tests indicated that a useful LCSR capability might be based on the structural information which had been extracted automatically. The final test performed combined the transition letter set information with a minimum amount of loop letter set information and a simplified measure of how typical the durations of component sounds were within the word "seven." It was shown that this combination of information was capable of discriminating correctly all sixteen real occurrences of the word "seven" in a balanced sample of 175 words taken from the LCSR vocabulary.

Summary of Phase 1

Rigorous Test of Structural Violation – A program was designed, written and coded for recognizing words using the transition and loop letter sets found in Phase 0. This program was exercised on all Interim Test data to investigate the frequency and nature of structural violations; i.e., the failure of expected sounds to occur, (a transition letter set violation) and the occurrence of completely unexpected sounds (a loop letter set violation). On 1042 words, transition letter sets were violated 5.3 percent of the time, and loop letter sets were violated 6.6 percent of the time.

Analysis of Structural Violations – An additional program was designed, coded and used to find the most plausible (usually the simplest) explanation for each of these structural violations. This revealed that the violations occurred in highly characteristic ways. For example, all violations of transition letter sets other than the last of a word were found to occur at critical positions, which were recognizable by the property that the violating feature was allowable in either the immediately previous or immediately following transition letter set. Furthermore, at most two features were violated within a transition letter set, and at most two transition letter sets were violated per word. These violation characteristics were interesting in their own right, as objective indications of coarticulation effects. (It was also noted that critical feature violations were more often due to the vocal gesture being affected by the gesture which follows than the gesture which preceeds, an effect reported by phoneticians.)

Recognition Algorithms which Accommodate Violations – In addition to their intrinsic interest, the limited and characteristic ways in which violations occurred were important, as they provided the opportunity to modify the recognition algorithm so it could accommodate structural violations while not admitting an impossibly large number of false recognitions.

Another simulation program was designed, coded and applied to determine the set of structural violations which should be accommodated to obtain a balance between false rejection and false recognition. In support of this activity a set of violation categories had to be formulated, and the rate of occurrence of each category, both of real and artifactual recognitions, had to be determined. This same program was used to gather data on the time durations of transition and loop sounds.

Dynamic Programming Formulation – Significant design changes to the mathematical machine recognition procedure were made to accommodate structural violations and to monitor transition and loop sound durations while keeping the number of copies of each machine at a minimum. In particular, the copy initialization, transition and dropping rules were modified so that the resulting algorithm can be interpreted as a method for solving a highly constrained maximization problem on recent preprocessor output history. Briefly stated, the function maximized is an additive quality measure for monotonic functions from an ordered set (the transition letter sets) into the recent preprocessor output. The class of functions admitted is constrained by the requirements that letters must either lie in the corresponding letter set or in a slightly larger set which includes the violations to be accommodated. Other constraints reflect the maximum number of transition and loop letter set violations, and the combinations that are allowed. The quality function reflects both the violation and a measure of typicality of the transition sounds. This dynamic programming interpretation of the word spotting algorithm makes much clearer the relationship between the mathematical machine approach and the work of Bridle[1].

---

1. W. Bezdel, J.S. Bridle, "Speech Recognition Using Zero-Crossing Measurements and Sequence Information," Proc. Inst. Elect. Eng., 1969, 116, 617-623.

Sound Duration Quality Measures – Programs were designed and coded to summarize the data gathered about the duration of transition and loop sounds. Based on these observations, models were created from which measures of how typical a given sound history can be derived. A modified Mahalanobis measure was derived in this way for transition sounds, and a totally different measure was established for loop sounds.

Real-Time Word Spotter – Using the information gained about violation categories and the modified (dynamic programming) recognition procedure, a real-time version of the word spotting algorithm was designed and coded, and named MEX (Machine EXerciser).

Word Interaction Model – The problem of discriminating between the real and artifactual recognitions produced by the word spotting portion of the recognition algorithm was analyzed, and a mathematical model was developed. (This mathematical model is described in detail in Section IV of this report, as the algorithm MINT cannot be understood without it.) The problem of selecting the words actually spoken is cast in this model as a problem in statistical decision theory.

Word Selection Algorithm – A dynamic programming solution was developed for solving the statistical decision theory problem which models word selection. The resulting algorithm is efficient both in execution time and storage. It is also flexible in that additional sources of knowledge can be incorporated into MINT as these additional sources become available in specific applications.

Statistical Models – In support of the statistical decision theory approach to word selection, statistical models were developed for each observable. These statistical models specify the distribution assumed for:

a. a priori real and artifact production rates

b. violation category occurrence rates

c. the transition sound quality measure

d. the loop sound quality measure

e. real/artifact association

f. initial delay, interword gaps and fixed delays.

Procedures were developed for extracting the statistical parameters of these distributions over Interim Test data.

Real Time Word Selection – A real time program was designed and coded using the dynamic programming solution to the word selection problem. The word selection program (MINT - Machine INTeraction), together with the word spotting program MEX, and interfacing programs constitute the program LISTEN.

LISTEN Tests – The completed system for real-time recognition of continuous speech was tested, first over Interim Test data to assure that the system was operating as intended, then over Test data. Results are reported in Section V.

Additional Speakers – The structural composition of three additional speaker's voice data were derived using the same methods applied during Phase 0 to the single voice. Structure was found for these additional speakers which was as rich as that found for the original speaker. The added speakers included a male with southern (Texas or Florida) accent, and a female of Chinese-American extraction. Full voice data were developed for the southern male (BRO) and LISTEN was tested for his Interim Test and Test data. Those results are also presented in Section V.

Analysis of Errors – In the limited time remaining after testing LISTEN with two speakers, some preliminary analyses were performed on the recognition errors observed on the original speaker's (MWG) voice. These results are also presented in Section V.

SECTION III

MEX, THE WORD-SPOTTING PORTION OF LISTEN

Development of the Algorithm

At the end of the Phase 0 of the LCSR project, we had demonstrated the existence of characteristic sound groups and time durations in speech data, and implemented programs for finding those sound groups. We had also shown that automatic recognition of continuous speech in real time was apparently feasible using an algorithm faithfully described in part by the non-deterministic finite transducer depicted in Figure 1. In this figure, the T_i are the transition letter-sets and the L_i are loop letter sets.

This transducer is non-deterministic only in its initial state S_0. In terms of simulation, this means that, whenever an incoming letter is in T_1,

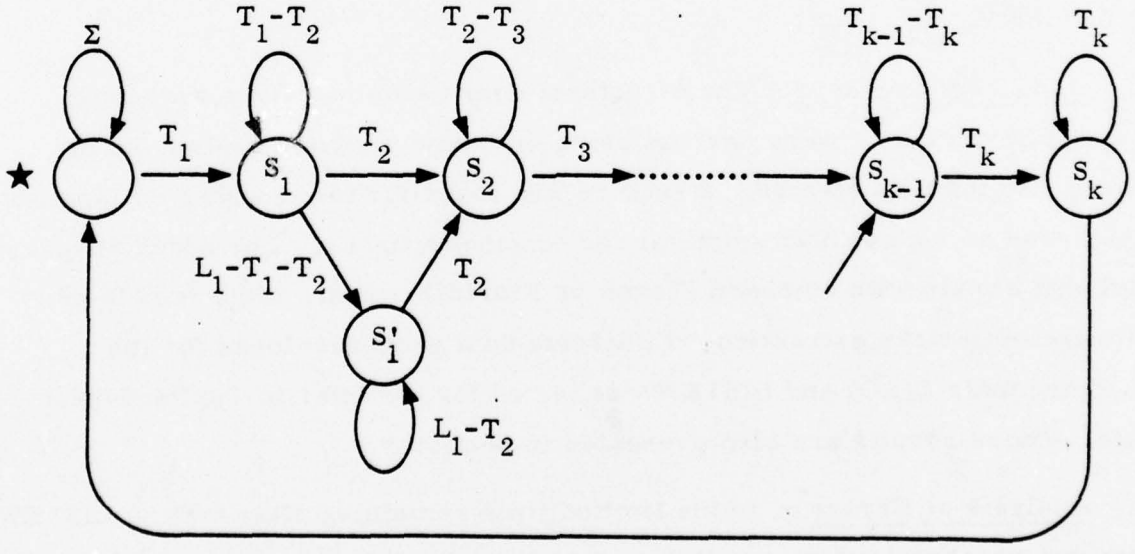


Figure 1. The Non-Deterministic Finite Transducer Developed in Phase 0

a new copy of the machine is started in state S_1, and allowed to follow the subsequent history of the incoming sound. The number of copies one had to deal with was reduced by introducing simple rules for (usually) dropping copies when, after transition, they end up in the same state.

We also had produced a program, REMEX, for exercising this algorithm on speech data. It was used to test the generality of the transition and loop letter sets on speech data other than those from which the T's and L's were derived. The detailed analysis of the results of this test on Interim Test data motivated the revised concept of machine operation to be presented here.

The non-deterministic finite transducer depicted above emits a potential recognition only for utterances which contain a subsequence of sounds (letters), corresponding to the sequence of transition letter-sets, and such that all of the sounds between these transition-causing letters are either in the associated transition or loop letter-set. As these letter sets were derived from training speech data in such a way that every example led to recognition, the subsequent test on Interim Test data gave the first reliable indication of how often speech examples will fail to have sounds from each of the transition letter sets, or exhibit inter-transition sounds not contained in the appropriate loop letter-set, or both. Failure of a word to meet these requirements for recognition by the transducer is called transition or loop letter-set violation.

Careful analysis of all examples which did not cause emission of the recognition symbol, accomplished with the aid of the computer program BADG, revealed important and interesting structural regularities in these violations. The revised machine concept takes advantage of these regularities to accommodate almost all of the observed violations while admitting the smallest possible number of false recognitions.

The Interim Test data contained 1042 words essentially uniformly distributed over the vocabulary, and with each vocabulary item occurring preceded by and followed by every vocabulary item. In this test, 6.6 percent

of all words exhibited only a loop letter-set violation, and 5.3 percent exhibited a transition letter-set violation.

Word-spotting is accomplished by modifying the finite transducer so as to accommodate (allow to reach the recognition state) as many as possible of these violating examples, while taking advantage of observed regularities among the violations to admit as few exceptional cases as possible.

Accommodating Loop Letter-Set Violations - Several regularities were observed in the 69 cases of loop letter-set (only) violations. For example:

1. Usually only one loop letter-set is violated per example. Of the 69 observed cases, 68 were violations of a single letter set and 1 was a violation of two loop letter-sets.

2. The number of letters which violate a given loop letter set, given that it is violated, tends to be small. Of the 69 cases, 49, 15, 4 and 1 cases involved violations by 1, 2, 3 and 4 letters, respectively

Equally important as the above observations is the fact that no solidly reliable criterion could be found for predicting which individual features of a loop letter-set were subject to violation and which are not. It therefore appears that the recognition procedure must accommodate arbitrary violations of the loop letter-sets, and provide for monitoring the number of loop letter-sets violated, and the number of letters which violate each loop letter set. A copy will then be dropped when the probability of a real example's exhibiting the observed violations becomes sufficiently small.

The modification to the machine structure which will accomplish this is obtained by adding two transition paths to each T-counter and L-counter state pair as shown in Figure 2 in dotted lines. When a machine copy transits via one of the dotted paths of Figure 2, the occurrence of an L-violation at the current state is noted.

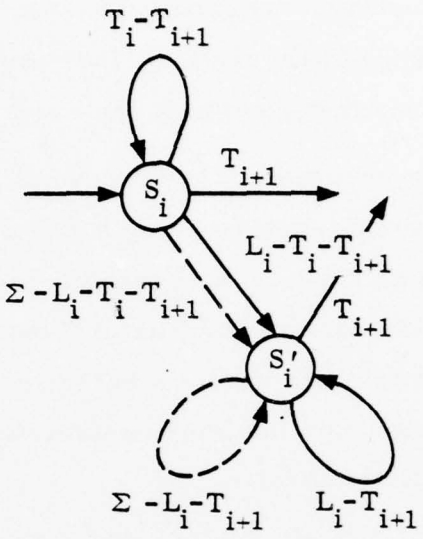


Figure 2. Modification of the Non-Deterministic Finite Transducer to Accommodate L-Set Violations

When comparing the modified and original machines, notice that the paths which have been added are traversed precisely when the original version machine copy would have been dropped. Dropping of copies is now associated with excessive accumulations of violations, rather than a single L-violation. Notice also that the modified machine is still deterministic at every state but the first; i.e., in each state (other than the first) there is precisely one transition path or action to be followed for any given input letter. Therefore, these added transition paths do not induce a need to produce new copies of the machine.

Accommodating Transition Letter-Set Violations - Analyzing the observed transition letter-set violations brought to light the concept of a critical feature position. Examining the sequence of transition letter-sets for any vocabulary item, it can be seen that each feature tends to go through an orderly evolution during the course of uttering the word. There tend to be

groups of several contiguous letter sets where the feature is reliably present or reliably absent, often separated by groups of letter sets which are indifferent to the presence or absence of the feature, indicating that the feature is more or less unpredictable in that portion of the word.

The regularity of this evolution is striking when first noticed. It suggests that the probability of a feature being set in a sample from the VIP-100 varies smoothly with time, in a repeatable fashion from voicing to voicing of a given word. When the probability of a feature being set becomes very high, the transition letter sets have 1's in that region. When the probability is very low, 0's occur, and for intermediate values, the transition letter sets are indifferent to the feature.

Viewing the 0's and 1's in the transition letter sets as arising from a smooth drift of probability during the evolution of the word, one might suspect that small variations in the word's development, and also statistical sampling effects, would cause those features near boundaries of regions of reliable occurrence or non-occurrence of the features to be the ones most vulnerable to violation. Thus, if a feature reliably does not occur in three consecutive transition letter sets, and may or may not be present in the next four transition letter sets, the probability of production of that feature is apparently rising, and is most likely to violate the last transition letter set in the contiguous string of transition letter sets which call for its absence. Features like these are considered to lie in critical positions. In general, a particular feature position in a particular transition letter set is considered to be a critical position if a 0 or a 1 is required to appear there, and its opposite value is allowed in either the preceding or following transition letter set. Thus, a feature position can be a critical position with respect to the preceding, or with the following transition letter set, or with respect to both.

Transition letter sets were violated in 55 (5.3 percent) examples of Interim Test data. Of these, 37 (3.6 percent) entailed violation of transition

letter set other than the last one in the word. All but one of these were violations of features in critical positions. This is very remarkable, as only about 30 percent of the violable features are in critical positions, but they account for 97 percent of the non-terminal transition letter set violations. These data indicated that, if the finite transducer were modified to accept violations of non-terminal transition letter sets when only critical features are violated, only one case (0.1 percent) of Interim Test data would be falsely rejected. It was reasonable to suspect that fewer false recognitions would be produced with this scheme than with another recognition technique which accepts a broader class of transition letter set violations. MEX takes advantage of this opportunity to accommodate violations observed in real recognitions. Furthermore, it was found that no cases involved violation of more than three critical features in any single transition letter set, and at most two examples showed violation of three critical features so MEX also rejects violation of more than two features in critical position of non-final transition letter sets.

When the final transition letter set of a word is violated (which occurs 1.8 percent of the time), it is typically not a violation of features which are critical with respect to the next-to-last transition letter set. It is instead at features which are critical with respect to the first transition letter set of the word which follows. (This explanation accounts for 16 of the 19 observed cases of final transitional letter set violations.) As there is no way to know which features are critical in this sense at the time the transition is made, the criticality criterion cannot be applied to limit the acceptance of final transition letter set violations. The alternative adopted is to accommodate only those violations of final transition letter sets which occur on two or fewer features.

In view of these observations, the modification of the finite transducer which is appropriate for investigating accommodation of transition letter set violations has the following characteristic. At each state of the machine,

a set of acceptable violations must be defined. These acceptable violations vary from state to state within the machine. Upon receipt of a letter, the machine should first check to see if the letter is in the normal transition letter set, say T_i. If not, the letter should be checked for membership in the set of acceptable violations, say T_i^*. When the letter is an acceptable violation, a new copy of the machine should be produced and advanced to the next state, unless the additional transition letter set violation would constitute a highly unlikely history for an example. The modification to a typical pair of states of the finite transducer is shown in Figure 3, with the new transitions shown dotted.

The transitions just added to the transducer make it non-deterministic at every state. The four previously existing transitions out of state S_i are labled with sets of letters which form a partition of the set of all letters (Σ). Therefore, if a letter is in the set of acceptable violations, $T_{i+1}^* - T_{i+1}$, it is also in one of the sets associated with a non-dotted transition.

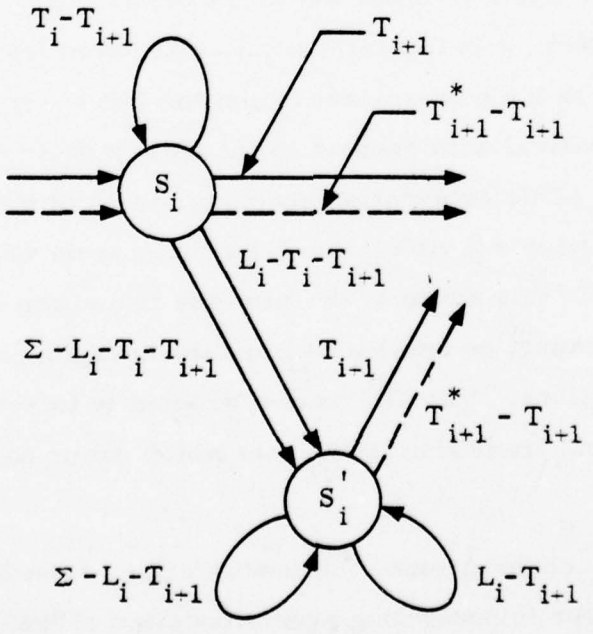


Figure 3. Modification of the Non-Deterministic Finite Transducer to Accommodate Transition Letter-Set Violations

Thus the old copy stays behind and is treated normally while the new copy is created and moved to the next state. The same situation holds for copies in state S_i'. This raises the spectre of a tremendous proliferation of machine copies.

Controlling Proliferation of Copies - In the finite transducer proposed during Phase 0 and implemented in the program REMEX, the number of machine copies active at any one time was kept low by sometimes dropping copies when they entered a state already occupied by another copy. The underlying rationale, still valid, is that only the most promising copies are of interest, and that of two copies in a given state only the one whose history is more typical of real recognitions need be retained. The selection and dropping criteria were implemented as coalescing rules, applied when a copy transited into an already-occupied state. The intention was that in the real-time recognition algorithm, all data accumulated about the copies' histories would be used in selecting the more promising one and which one to drop.

There are two difficulties with choosing which copies to drop and which to save at the instant a copy moves into a new state. Both have to do with the fact that it is impossible to determine at that moment which of the two copies is more promising or more typical. First, assume that there is a measure of the quality of a machine history which depends upon the number, type and accumulated counts of letters in each state, and that the measure is non-negative and additive on states: that is, for each state S_i there is a non-negative quality measure q_i which depends only on L_i, the set of letters and their counts processed in state S_i, and the quality of a machine-history is the sum of the $q_i(L_i)$ over all the states. Suppose copy C transits into state S which is already occupied by copy C'. The histories of both copies is complete on states preceding S, and the contribution of those states to each copy's

history quality can be computed. But since C and C' move into state S at different times, the times they have been in state S will be different. So this contribution to their history qualities is not known, and may be different when it finally is computable. So, which copy will ultimately have the better history is not decidable at the instant C moves into the new state.

Consider, on the other hand, the instant at which a letter in T_{i+1} or T^*_{i+1} arrives. All copies in state S_i are then candidates for transition to the next state. Since the qualification to transit does not depend upon the history of copies, but only on the letters received and the current state, the subsequent history of all copies advanced out of state S_i will be identical, and since the quality measure is additive, the copy (among those in state S_i) which will ultimately have the most promising history is the one with the most promising history now. Therefore, the most promising copy can be selected among copies transiting out of a given state, but a copy transiting into a state cannot be compared with those already in a state. Instead of coalescing rules, one needs selection rules. More importantly, all copies transiting out of a given state should be processed simultaneously, and only the most promising one should be allowed to move forward. This new way of organizing and looking at machine operation reveals that the machine is now really a mechanism for doing dynamic programming on machine histories!

We have just described one difficulty with coalescing rules, and out of that difficulty extracted a better way of organizing and viewing machine operation. But there is a second difficulty to consider which is not as felicitous. As will be described in connection with interaction of machines, a very useful measure of the quality of the history of a given machine is the conditional probability that the actually observed history would arise, given that the word is actually being spoken. Since we are interested in selecting the most promising history, we may use the negative logarithm of this probability and choose the minimum value rather than the maximum. The difficulty is that

this is not an additive function on machine states, unless the events in one state are independent of the events in another state.

As an example, consider the T counter values which are observed in various states. The Mahalanobis metric computed from these counter values (as described in the Phase 0 final report) is linearly related to the logarithm of the conditional probability that the observed counter values can arise, given that the word is really being spoken. This is one component of the desired measure of quality of a machine's history. But the dispersion matrices used in computing the Mahalanobis metric have (in general) non-zero off-diagonal elements, so it may include a term proportional to the product of the deviation of two counters, say the i^{th} and the j^{th}, from their median values. The value of this contribution cannot be computed until the later of states i and j is exited, so in principle at least, it is impossible to select the most promising copy upon exit from any state which is followed by a state whose counter values are correlated with counter values from the current or a previous state. This example clearly shows the relation between additivity of the quality function, independence of events in various states, and selection of most promising copies. If the counter values accumulated in two states are statistically independent, the corresponding off-diagonal element in the dispersion matrix is zero and the problem mentioned above does not arise.

What can be done about the lack of independence among states and rules for selecting copies? The only practical answer is to ignore the problem, and postulate an additive quality measure. The result is that the copy which will be selected is not guaranteed to be the one which ultimately will have the best history, but rather the one which, assuming that all subsequent behavior is normal, will have the best history.

General Description of Dynamic Programming Machine Operation - Following is the description of a mode of machine operation which implements

the ideas put forth above. For its operation, we assume that each machine copy carries with it a summary of its history, and that there is a function (described later) which assigns a quality measure to that history when the copy transits to a T-counter state. The historical summary carried with the copy can contain data about T and L counter values and violations of any kind. We also assume there is a rule (also described later) for determining when a copy should be dropped due to too many violations of any sort. The quality of a copy means the quality of the copy's associated history.

We divide the machine into stages, consisting of pairs of T and L counter states, and describe the necessary processing for all copies at a typical stage; i.e., in either state of the stage. Stages are processed from the last to the first (backwards along the machine). Processing of the last stage and the initial stage are described later. A typical stage is shown in Figure 3.

All of the copies at a given stage are processed as a group. The processing depends primarily on the incoming letter and is described by the following rules:

a. Incoming letter in T_{i+1}

The quality of each copy at this stage (both T and L states) is computed after updating its history to indicate termination of the current stage. The copy with the highest quality is advanced to the T state of the following stage. All remaining copies are dropped.

b. **Incoming letter in $T^*_{i+1} - T_{i+1}$**

A potential new history is computed for each copy at this stage by assuming that the copy will be moved to the next stage at the cost of a transition letter-set violation. If this potential history meets the dropping criteria, it is not considered further. If it does not, its quality is computed. The potential history of the highest quality is selected, and a new copy is created in the T state of the next

stage and given that history but only if it has potential of being selected out of the next state. If none of the potential histories survives the dropping criteria, no new copy is created. All copies at this stage are also processed according to whichever of the following rules applies.

1. Copy in State S_i, Incoming letter in $T_i - T_{i+1}$.

   The copy remains in this state.

2. Copy in State S_i, incoming letter in $L_i - T_i - T_{i+1}$.

   The copy's history is updated to indicate termination of the T_i state, and assigned to the S_i' state.

3. Copy in State S_i, incoming letter in $\Sigma - L_i - T_i - T_{i+1}$.

   The copy's history is updated to indicate termination of the T_i state and a loop letter-set violation. If the copy meets the dropping criteria, it is dropped; otherwise it is assigned to state S_i'.

4. Copy in State S_i', incoming letter in $L_i - T_{i+1}$.

   The copy remains in this state.

5. Copy in State S_i', incoming letter in $\Sigma - L_i - T_{i+1}$.

   The copy's history is updated to indicate violation of this loop letter-set. If the copy meets the dropping criteria, it is dropped; otherwise it remains in this state.

Copy Initialization -- Violation of the first transition letter set of a word was found to occur rarely, and then only at the beginning on an utterance. However, when the first transition letter set was violated, it did not occur in a feature in critical position, and did not occur in more than two features. These observations were incorporated by providing that MEX start new copies only when the incoming letter was a member of the first transition letter set, except upon receipt of the first letter of an utterance. For that letter only, violations of up to two non-critical features are considered potential conditions for starting a new copy.

Additional conditions were imposed on the starting of new copies to reduce the total number of copies typically active, and thus reduce the computational burden in MEX. The additional rules are:

a. No new copy is started less than 25 time counts (25 x 2.2 = 55 milliseconds) after starting a copy without violation of T_i.

b. No new copy is started in stage 1 with a violation if there is already a copy there without violation.

The circumstance necessitating rule a is long sequences of contiguous incoming letters, all lying in T_1. This happens, for instance, with MWG's enunciation of the words "six," "seven" and "zero." Unrestrained production of machines when these words are spoken would lead to a pile-up of many, nearly identical, copies at Stage 2 or 3, in State T. On the order of twenty valid start letters (i.e., letters in T_1) occur quite often. These machine copies may not accumulate in Stage 1 because, for the machines mentioned, T_2 and T_3 contain or are equal to their predecessors, or otherwise tend to be traversed very quickly. In general, the machines do accumulate at the first stage whose transition letter-set is significantly different from T_1.

Final Stage Processing -- Processing in the final stage of the dynamic programming machine must also be somewhat different from the previous machine versions. The older version included a looping transition on letters of the last transition letter-set, included to accumulate a count of the number of contiguous letters in the last transition letter-set, in the hope that the additional count would provide some additional power for discriminating between real and false recognitions. This organization of the last stage is shown in Figure 4 (a).

When the machines are modified to accommodate transition letter-set violations, it is important to include provision for violation of the last transition letter-set, as Interim Test data show that about 1.8 percent of the examples violate the last transition letter-set and about 3.5 percent of the words violate transition letter-sets other than the last. A terminal transition letter-set is apparently about four and one-half times as likely to be violated as a non-terminal one. If the last stage of a machine is entered via a transition letter-set violation, it makes little sense to try to count the ensuing number of contiguous occurrences of letters in the final transition letter-set. If there are any such letters at all, there is no need to transit into the last stage via a violation. So last-stage counting and accommodating violations are interrelated. The 1.8 percent of cases which can only reach the recognition state via violation of the last transition letter-set will necessarily have a final count value of zero. Statistical tests using the last count value would therefore have to account for the dependence between violation and count values. We thus rationalize dropping the final stage counter.

If, however, we let the machine emit a potential recognition upon receipt of each letter in $T_k^* - T_k$, it would frequently occur that a whole sequence of potential recognitions would be emitted with violations at the last transition letter-set, followed immediately by the real recognition, with a higher count value in the next-to-last stage, and no violation at the last transition. Since these extraneous potential recognitions would be emitted in a brief time compared with the whole length of the word, it is clear that a short delay would be sufficient to eliminate them. We therefore introduce a fixed time delay (25 counts, or 55 milliseconds) between entering the last stage and emission of a potential recognition, as shown in Figure 4 (b). During the stay of a copy in the delay pipeline, we apply a selection criterion, allowing only the single most promising copy to survive. This eliminates almost all of the extraneous recognitions, but admit the 1.8 percent of all cases which can only be accommodated by allowing violation of the final transition letter set.



(a) Unmodified form showing loop transition for counting letters in last stage.

(b) Modified form to accommodate transition letter-set violation with minimum extraneous emissions.

Figure 4. Unmodified and Modified Last Stage Processing

Dropping Criteria -- The rate of producing false recognitions is kept as low as possible by dropping copies which have experienced too strange a history to be plausible as real recognitions. Criteria for determining what constitutes too strange a history were developed empirically. To this end a research version of the dynamic programming machine algorithm was implemented, and a set of acceptable violations were determined that accommodate all but 3 examples (0.3 percent) in the Interim Test Data set (for MWG).

The criterion used in developing these rules was to select that set of violations which accommodated the maximum possible number of examples in Interim Test data, and which exhibited the highest ratio of real to artifactual recognitions. In the process of developing these rules it was found that many explanations for violations had been overlooked in the original analysis of these cases by the program BADG, reflecting the bias towards accepting violations of a particular category inherent in that program. For example it was initially thought that five cases of violations involved letters which violated a loop letter set three or four times. The more sophisticated dynamic programming machine simulation found alternative violations for these five cases which were more typical (they involved the most common type of transition letter set violation). Therefore, violation of more than two loop letter sets could be discarded as exceptionally atypical.

The dropping criteria resulting from this empirical study can be summarized in terms of the following categories of violations:

a. Loop Letter Set Violation Types

LI Violation of one loop letter set by one letter.
LII Violation of one loop letter set by two letters.

b. Transition Letter Set Violation Types

TI Violation of one non-initial, non-final transition letter set at one or two critical feature positions.
TII Violation of the initial transition letter set by the first letter of an utterance, at one or two (not necessarily critical) feature positions.
TIII Two violations of type TII
TIV Violation of the final transition letter set at one or two (not necessarily critical) feature positions.

In terms of these categories of violations, the dropping criterion used in MEX specifies dropping (or not creating) a copy if its history entails

a. any violation other than those listed, or

b. any combination of listed violations other than TI together with LI or TI together with LII.

Quality Functions -- Three aspects of a machine copy's history are observed by MEX. They are:

a. Transition and loop letter set violations

b. Letter counts while in T-type states (the state first entered at each stage)

c. Letter counts while in L-type states (states entered only when an incoming letter does not induce transition to the next stage, or remaining in a T-type state)

The first two of these aspects are considered when several copies at a given stage are compared to determine which one (if any) should transit to the next stage when the incoming letter is in the transition letter set, or which copy should spawn an offspring.

The admissible types and combinations of transition and loop letter set violations have been assigned category numbers in increasing order of rarity of occurrence. Thus the higher the category of a copy's violation history, the more atypical that history is, as measured by the ratio of its probability of occurrence for real recognitions to its probability of occurrence for artifactual recognitions. The quality categories for allowed violation types are given in Table 1 in terms of the violation types defined previously.

The transition state letter counts are used to compute a transition letter count time history quality function, denoted Q_T, which is related to the modified Mahalonobis distance function described in the Phase 0 final report. Q_T is a non-negative additive function which is small for typical histories and large for atypical histories. It is defined as follows.

Let

$Q_T(S)$ be the value assigned to Q_T for a copy exiting stage S

$Q_T(S-1)$ be the previously assigned value of Q_T for that copy

C_S be the letter count accumulated by the copy while in stage S.

M_S, k_S^+, k_S^- be constants associated with stage S of the machine.

Then

$$Q_T(0) = 0$$

and

$$Q_T(S) = Q_T(S-1) + (C_S - M_S)^2 \, k_S, \text{ where } k_S = \begin{cases} k_S^+ \text{ if } C_S \geq M_S \\ k_S^- \text{ if } C_S < M_S. \end{cases}$$

The constants M, k^+ and k^- have the following significance. Let M_S be the median of the distribution of T-state counts observed in stage S. For each

TABLE 1. QUALITY CATEGORY

		Type of Loop Letter Set Violations		
		None	LI	LII
Type of Transition Letter Set Violation	None	0	1	4
	TI	3	5	8
	TII	6		
	TIII	7	Not Allowed	
	TIV	2		

stage define a measure of deviation, a_S, to be associated with an observed count value C_S;

$$a_S = \frac{C_S - M_S}{W_S}, \text{ where } W_S = \begin{cases} W_S^+ \text{ if } C_S \geq M_S \\ W_S^- \text{ if } C_S < M_S \end{cases}$$

Here W^+ and W^- are chosen to be the mean of positive and negative deviations from the median, respectively. (This transformation is chosen to produce a more symmetric distribution than that exhibited by the raw count values.) Then

$$k_S^+ = d_S/(2W_S^+)^2$$

and

$$k_S^- = d_S/(2W_S^-)^2$$

where d_S is the value at location (S, S) in the inverse of the covariance matrix of a_S.

To select a copy for advancement or to spawn an offspring, that copy is chosen which has minimum Q_T value, among those with minimum quality category. Thus violation history is considered first, and among those copies with equal quality category, the one with most typical T-counter history is chosen.

The L-counter quality function, Q_L, which reflects how typical a copy's L-counter history is, is analogous in several respects to the T-counter quality function, Q_T. It is non-negative, additive, and larger for less typical L-counter histories. It is defined as follows.

Let

$Q_L(S)$ be the value assigned to Q_L for a copy exiting stage S

$Q_L(S-1)$ be the value previously assigned to Q_L for that copy

C_S be the letter count accumulated by the copy while in stage S

k_S, λ_S be constants associated with stage S of the machine.

Then

$$Q_L(0) = 0$$

$$Q_L(S) = Q_L(S-1) + k, \text{ where } k = \begin{cases} 0 \text{ if } C_S = 0 \\ \lambda_S(C_S-2) + k_S \text{ if } C_S > 0 \end{cases}$$

The significance of Q_L is as follows. Assume that the L-counter value has a probability P_o of being zero, and that its probability of being positive is zero at one, and exponentially decreasing for values greater than or equal to two. Then λ is the parameter of the exponential distribution, and Q_L is the negative natural logarithm of the ratio of the probability of occurrence of all the observed L-counter values to the probability they will all have value zero, assuming counter values at each stage are mutually independent.

Description of Algorithm

Figures 5 through 15 are flowcharts of the MEX algorithm, describing the word-spotting processing at a fairly high level. These flowcharts show several unique characteristics imposed by the need to minimize, as much as possible, the computational burden in MEX.

Input – MEX receives, from a peripheral device handler, pairs of computer words containing the VIP-100 features 17 through 31 (called the incoming letter), and an integer equal to the number of repeated occurrences of the incoming letter (called the letter count). The peripheral device handler ignores any isolated occurrence of a letter, so the letter count is always two or more, and successive letters are always different. Letters arrive at the input to MEX at intervals not less than 4.4 millisecond, and on the average every 13.5 milliseconds.

Output – The output of MEX is a notification, sent to MINT, of a potential recognition. The information sent to MINT includes the vocabulary item, the time of the beginning and recognition of the word and observations about the recognition, including the quality category (reflecting any structural violations), and the T-counter and L-counter history quality functions, Q_T and Q_L.

Processing – Each vocabulary item has one or two associated machine types (for initial and non-initial versions of the transition letter set characterization of some vocabulary items, e.g. MWG's "two"). Each machine type has several (five to fifteen) stages. Each stage, except the last, has a T-state and an L-state, and there may be several copies of the machine active in each state. For every incoming letter the processing problem is therefore, to determine its impact on each copy, modify that copy as appropriate, and to produce new copies when appropriate. This processing is described in the flow charts as if each copy moves from state to state through

the machine structure. In reality a copy is represented as a data packet in a singly-linked list, and movement consists of changing data (sometimes descriptive data and sometimes links) in data packets.

In these flowcharts, a rectangular box with double ends indicates a procedure described on another (usually following) page.

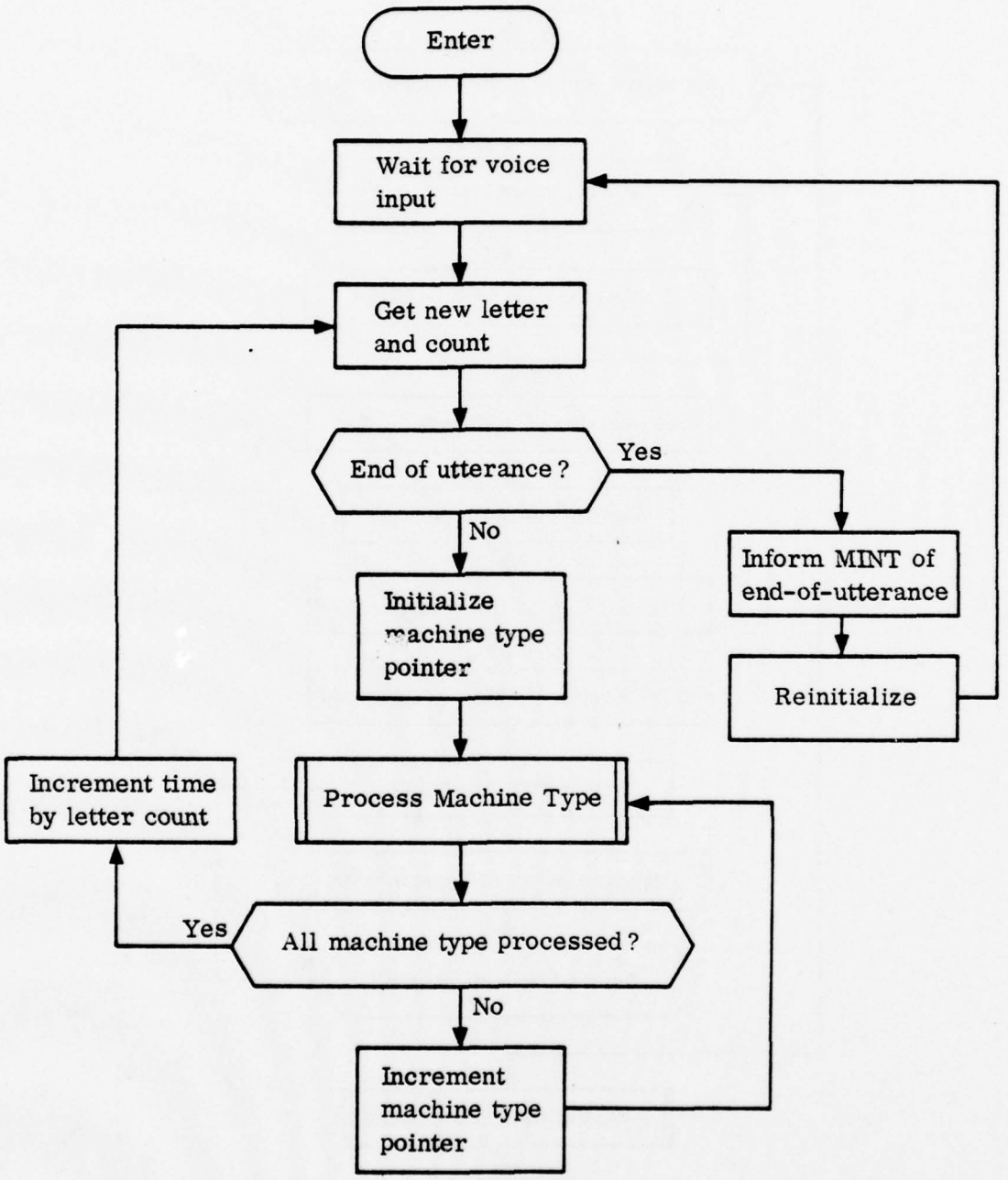


Figure 5. Outer Loop

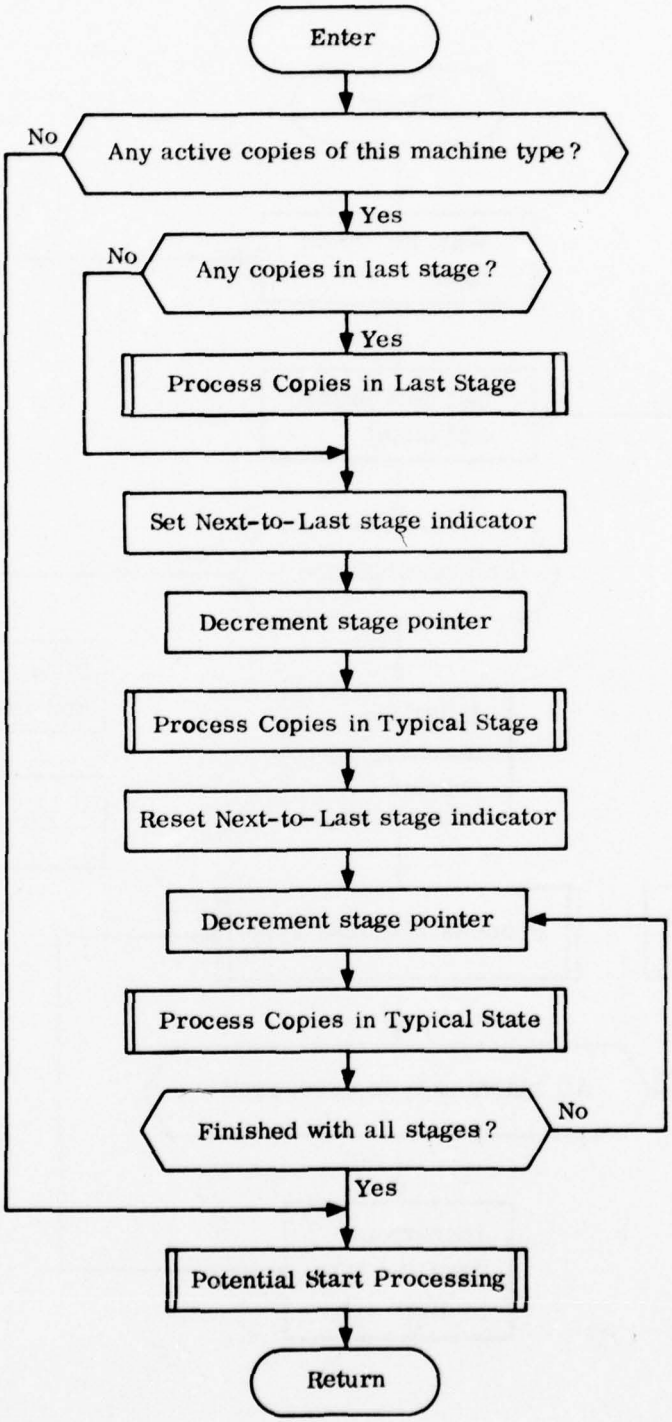


Figure 6. Process Machine Type

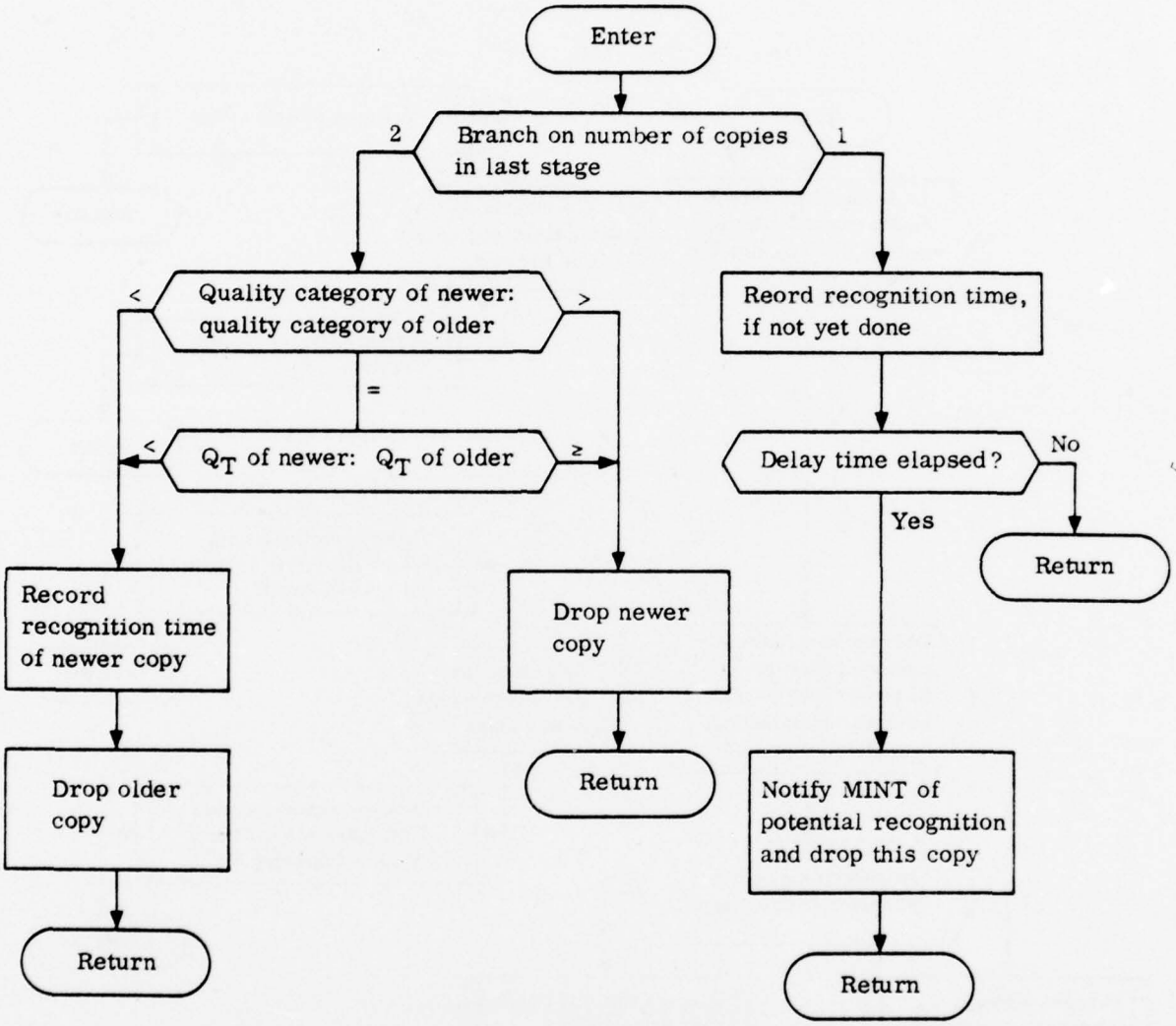


Figure 7. Process Copies in Last Stage

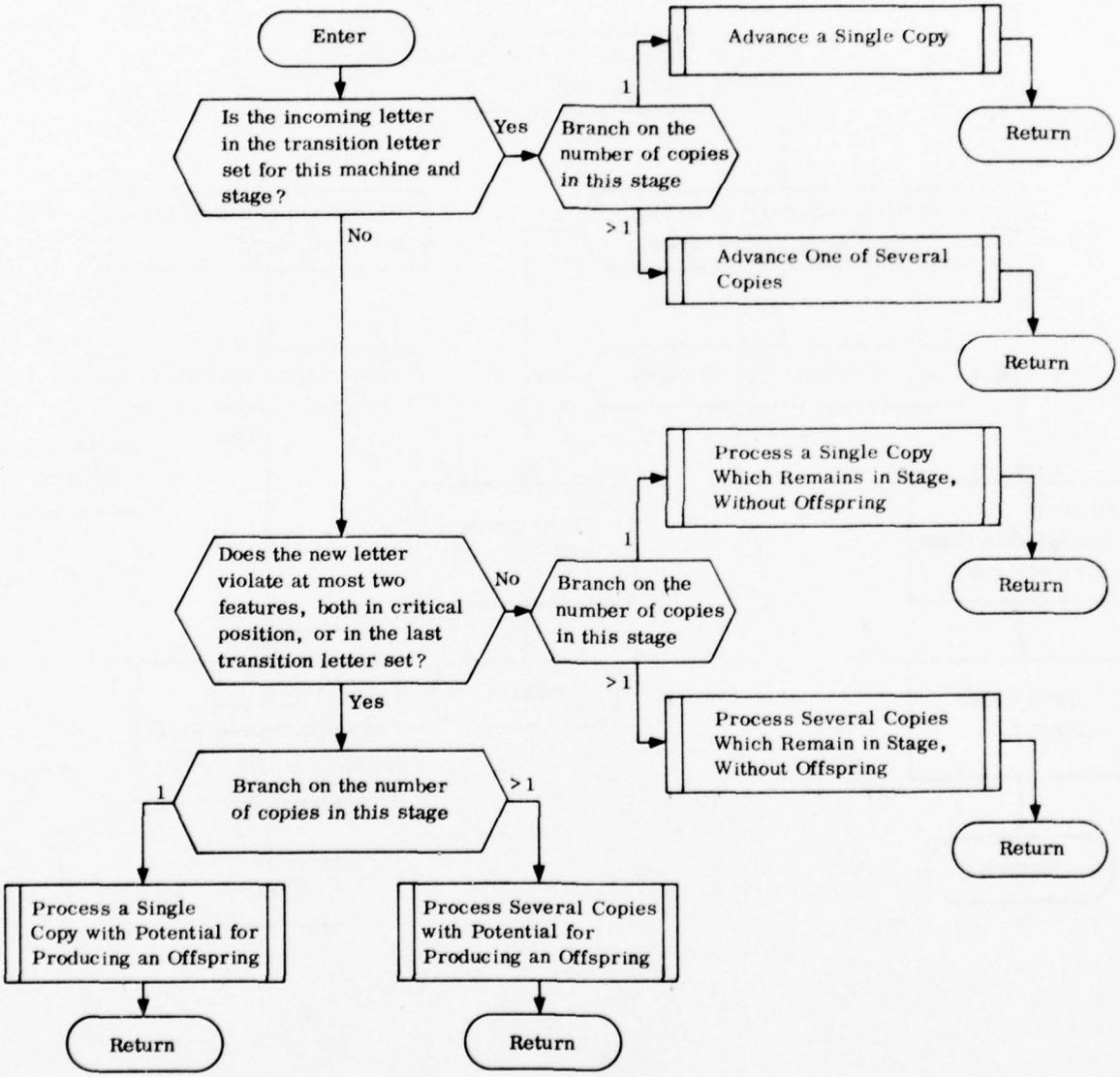


Figure 8. Process Copies in Typical Stage

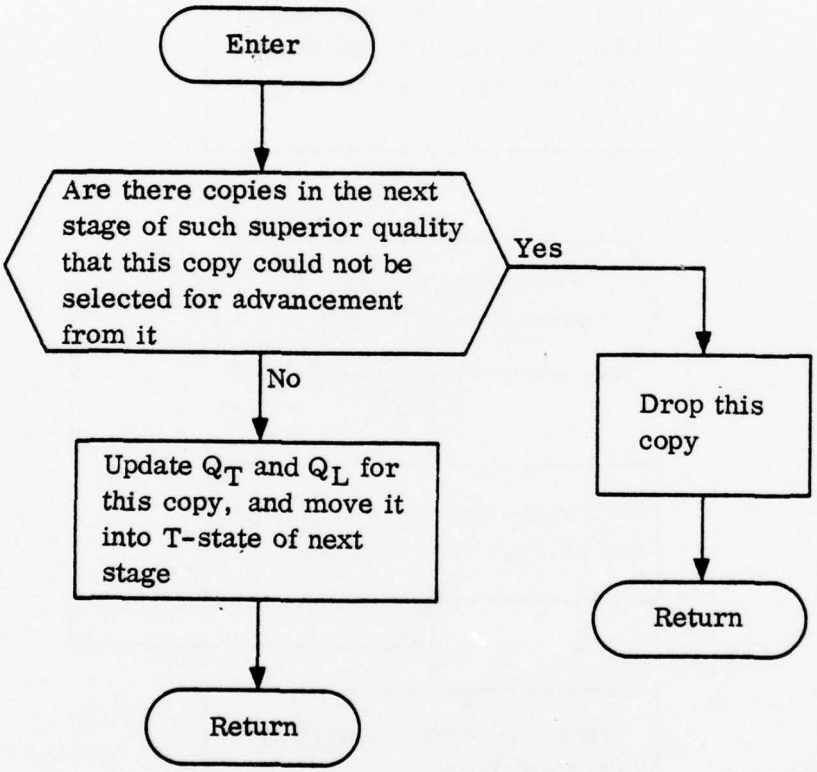


Figure 9. Advance a Single Copy

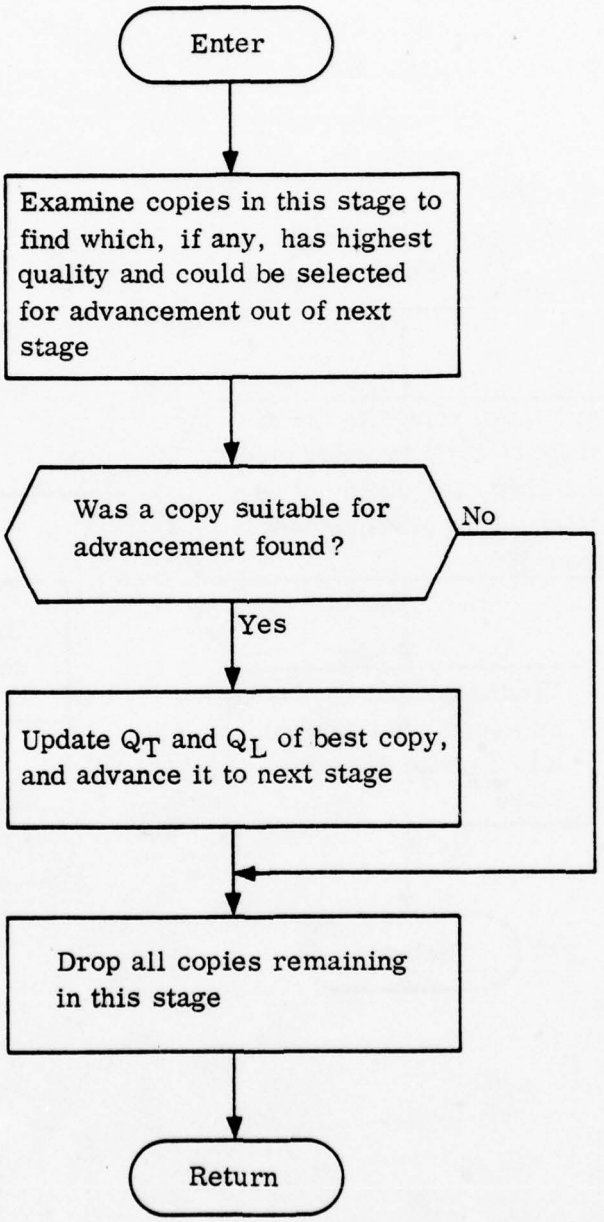


Figure 10. Advance One of Several Copies

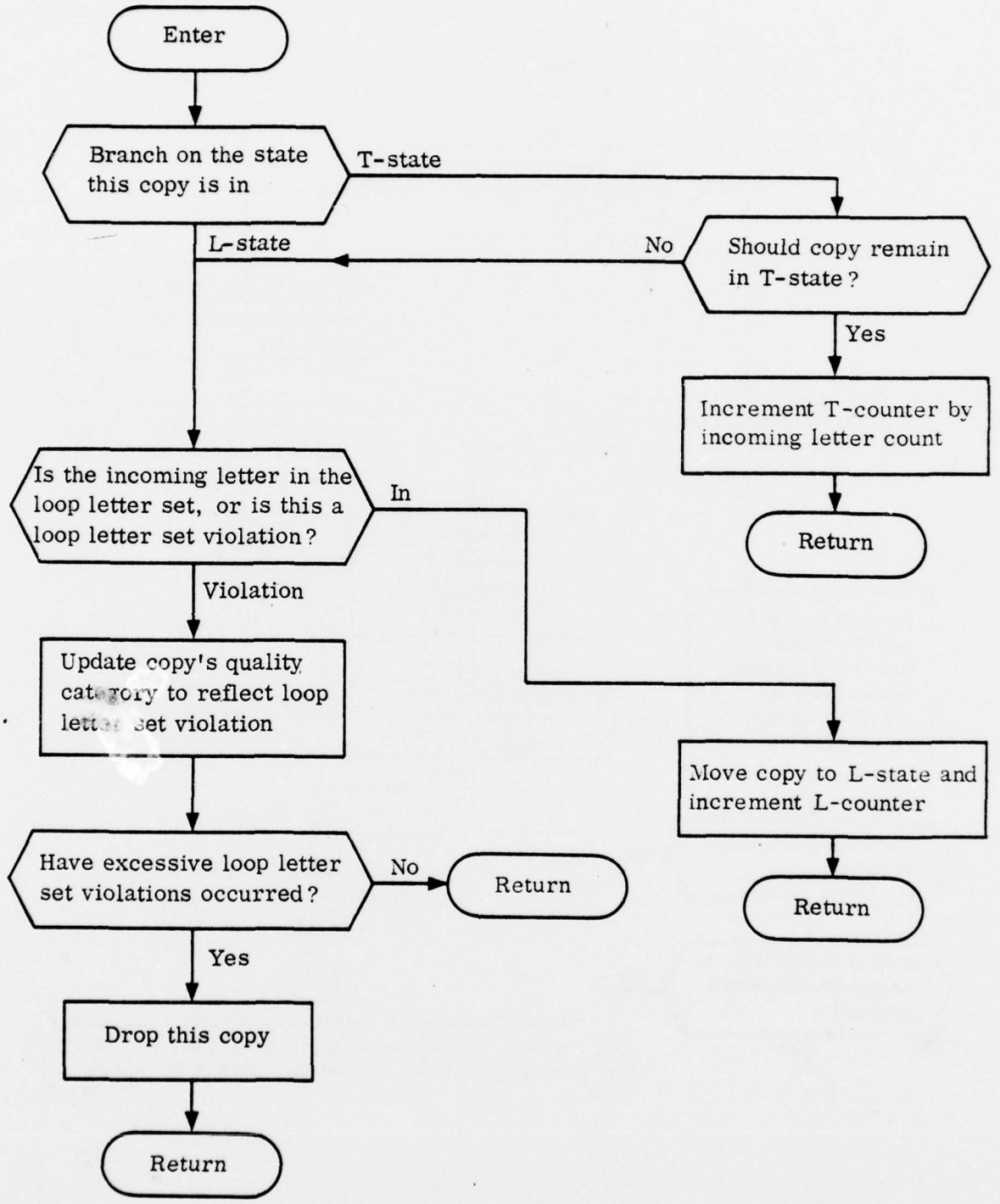


Figure 11. Process a Single Copy which Remains in Stage without Offspring

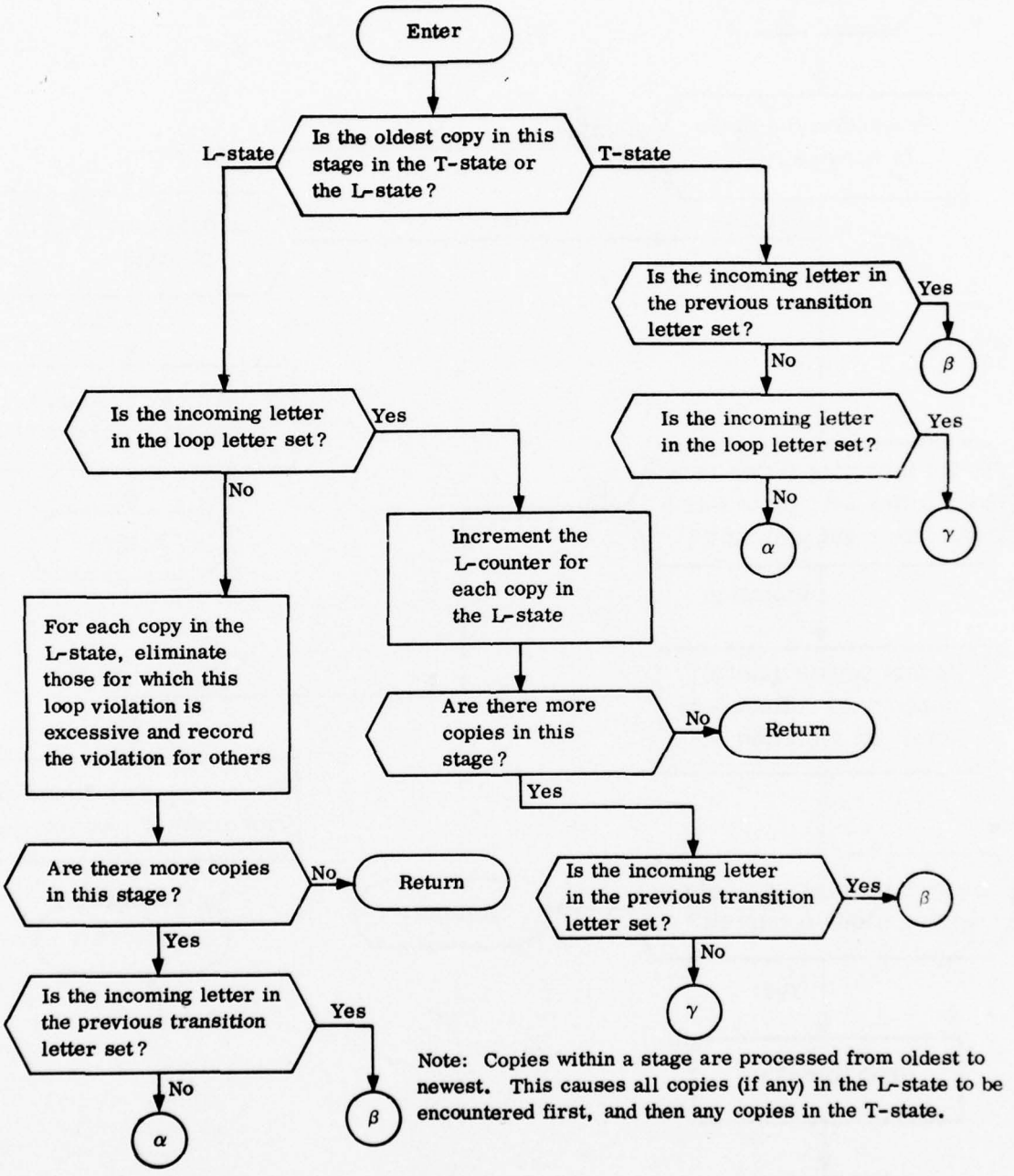
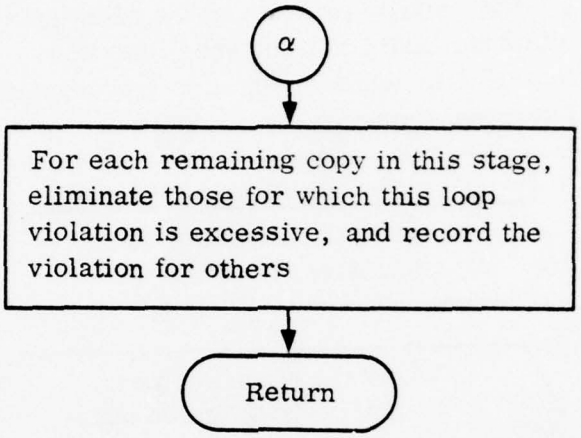
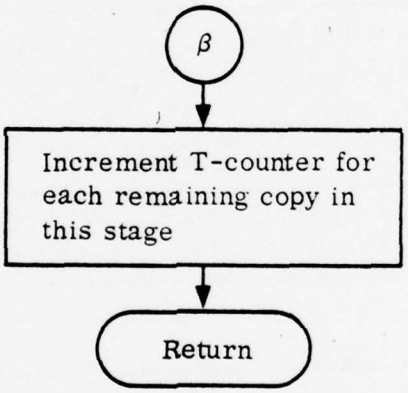


Figure 12. Process Several Copies which Remain in Stage, without Offspring

α : Processing copies in the T-state moving into the L-state via violation, or dropped due to excessive violation.



β: Process copies in the T-state which remain in the T-state.



γ: Process copies in the T-state which move to the L-state without violation

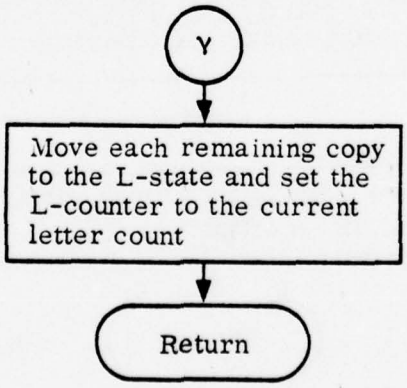


Figure 12. Process Several Copies which Remain in Stage, without Offspring (Cont'd)

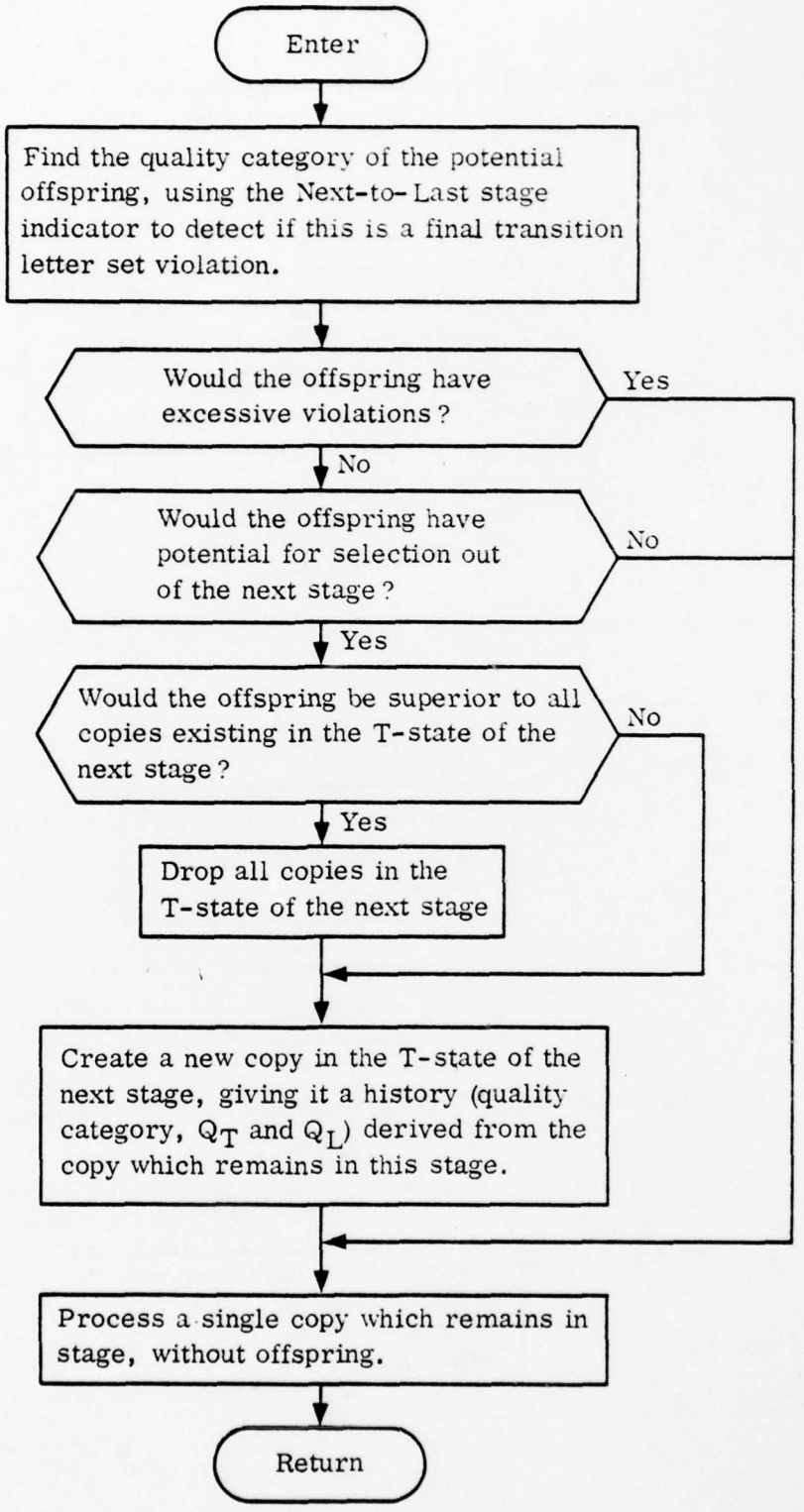


Figure 13. Process a Single Copy with Potential for Producing an Offspring

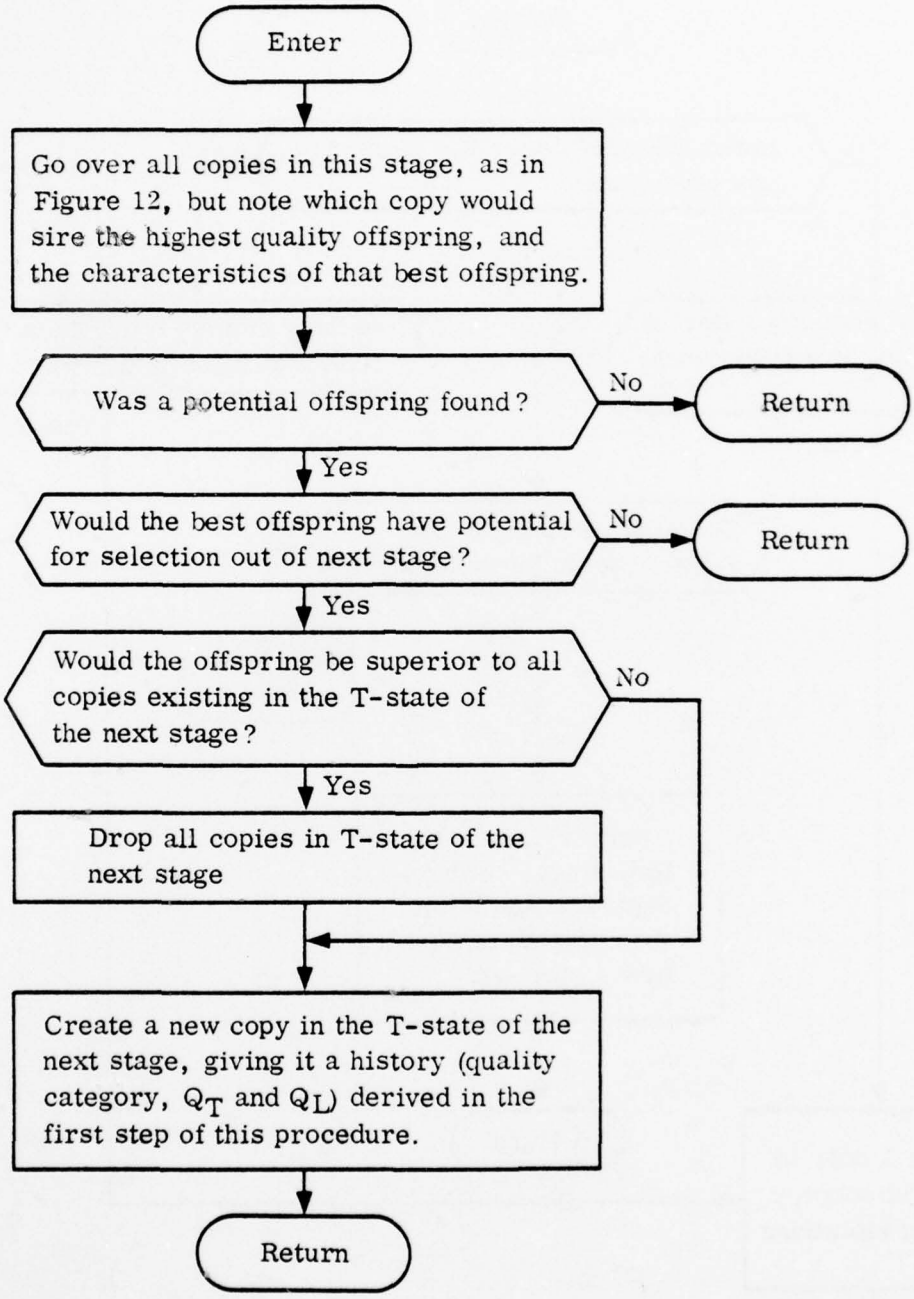


Figure 14. Process Several Copies with Potential for Producing Offspring

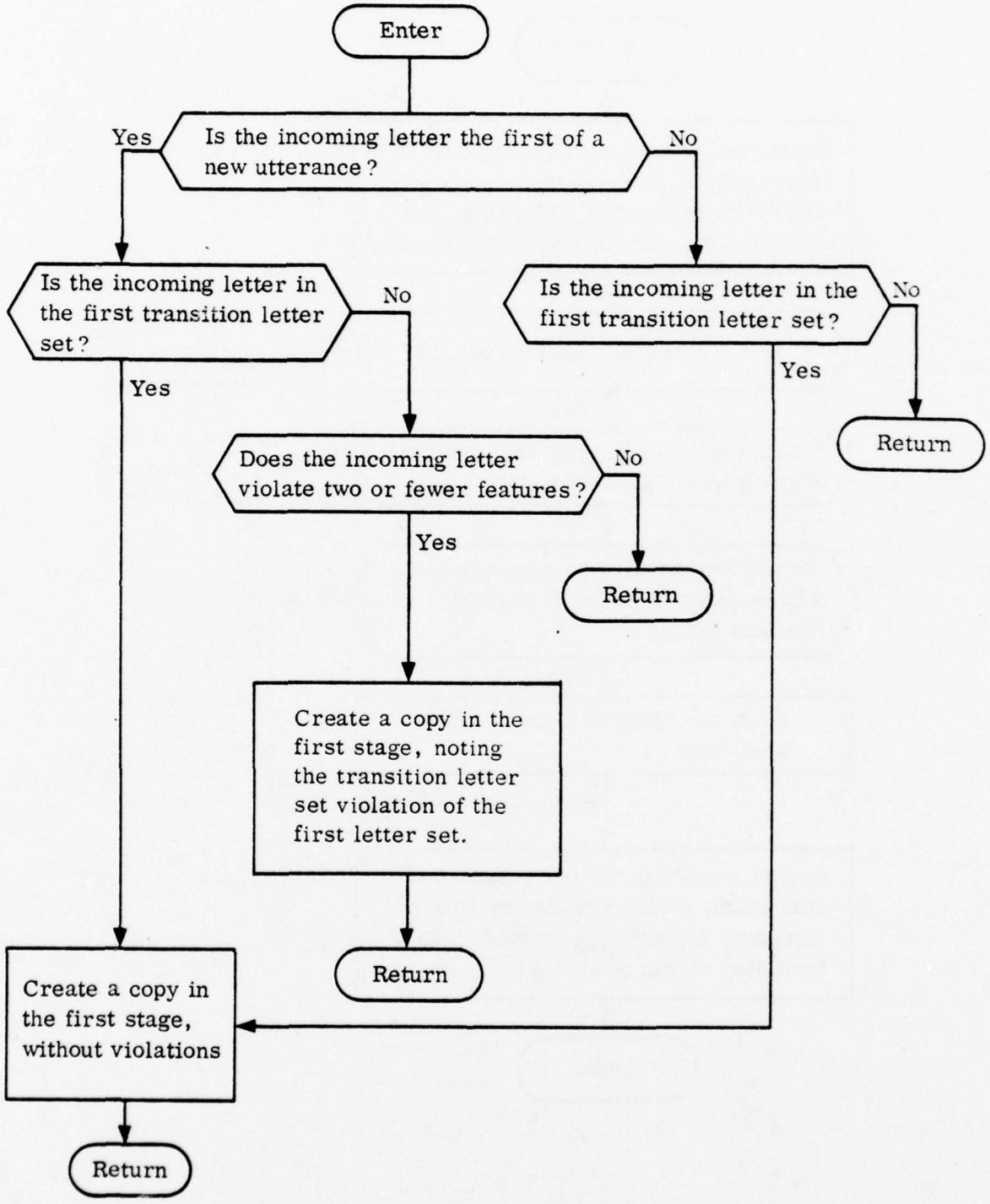


Figure 15. Potential Start Processing

SECTION IV

MINT, THE WORD SELECTING PORTION OF LISTEN

The Word Interaction Problem

MEX, the word-spotting portion of LISTEN, is concerned with detecting the presence of individual vocabulary items in the incoming stream of sound. To do this, MEX considers only the presence or absence of sounds typical of individual words, and operates as several parallel but entirely independent word spotters. The output of MEX therefore consists of notification of the potential detection of vocabulary items, information summarizing how typical the observed sound stream was for that vocabulary item, and the start and end times of that portion of the sound stream containing the suspected vocabulary item. This output consists of a combination of real recognitions and artifacts, or false recognitions, which must be eliminated in MINT.

The MEX-MINT combination is a purely serial process. MEX's identification of potential recognition is not influenced in any way by knowledge obtained about the utterance in MINT, and MINT does not seek an explanation of the utterance except by selecting some of the potential recognitions supplied by MEX. The rate of rejection of real recognitions which can be attained by the combination is therefore no better than the rejection rate achieved by MEX. MEX must therefore err on the side of acceptance rather than on the side of rejection, and its output will inevitably be rich in artifacts. (LISTEN operates with about three artifacts for each real recognition supplied to MINT.) Fortunately, simple observations about the temporal relationship of potential recognitions supply a rich information source for discriminating between real recognitions and artifacts in MINT.

Observables

We distinguish two classes of observations about the incoming utterance. First are the observations which are made within each individual recognizing machine, without regard to where in the utterance the contributing sounds occur and without regard to what any other recognizing machine may be doing. These we call the intrinsic observables, and we shall not stipulate (until later) what they are. Using statistical models of these observables it is possible to estimate the conditional probabilities that the observed values and characters would occur, given that the sounds causing the recognition are a real occurrence of the word, and given that the sounds are from other words; i.e., that the recognition is in fact an artifact. These conditional probabilities summarize the important things that can be inferred about the individual recognitions, without regard to relationships among recognitions or the timing of a recognition within the entire utterance.

Second are the observables which relate a given recognition to other recognitions and to the whole utterance. We will concern ourselves (initially, at least) only with interactions which reveal themselves through the start and end time noted during individual recognitions. For the purpose of studying interactions among potential recognitions we may therefore regard individual recognitions as characterized by three factors:

a. As real or artifact

b. By start and end time within the utterance

c. By two conditional probabilities that the intrinsic observables occur given that the recognition is real, and given that it is an artifact.

Our problem is, essentially, to infer the first characteristic of a set of recognitions, given the second and third for each. The order of speaking the real words thus identified is a relatively trivial problem.

Key Relationships - Two relationships among recognition and the whole utterance have been found especially useful. These are the delay or overlap between recognition or between a recognition and the beginning or end of the utterance, and the coincidence of real and artifactual recognitions.

Examination of many utterances reveals that the time between the beginning of an utterance and the start time of the recognition of the first (real) word of the utterance (called the start delay) is quite regular. It tends more or less often to be zero, and the non-zero cases have a characteristic distribution. Both the probability of a zero value and the mean of the non-zero cases are characteristic for each vocabulary item. Artifacts also have a concentrated tendency to have zero start delay, but have a very diffuse distribution of non-zero start delays.

The inter-word time delays (or time of coincidence) between the end time of one recognition and the start time of the following real recognition also is quite characteristic of the leading and following vocabulary items. Artifacts can often be eliminated as potential real recognitions simply because they exhibit inter-word delays which are unusual among real recognitions.

Many utterances have been observed wherein the real recognitions can correctly be identified and separated from artifacts simply on the basis of start delay and inter-word delays. One simply cannot construct a realistic path from the beginning of the utterance to its end using any recognitions other than the real ones.

It is not surprising that end delay, analogous to start delay, can contribute to the discrimination between real recognitions and artifacts. MINT should therefore make use of this observation in much the same way that it treats start delay.

Another regularity which can be noticed in real speech data is the occurrence of artifacts of one vocabulary type in association with real recognition of a word of the same or another vocabulary type. It has been noticed before, for example, that the words "nine" and "five" tend to be associated. The most fortunate feature of this type of regularity is its asymmetry - if a word of one type frequently causes an artifact of some other type, the converse is not at all necessarily true. Thus, when "nine" and "five" occur together it may be a very strong indication that "nine" is the word spoken. Because of this asymmetry, some artifacts can actually contribute to proper unravelling of a complex of recognitions.

Artifacts are frequently observed that overlap two contiguous real recognitions. These cases bring one's attention to the problem of establishing the criteria used to determine when two recognitions are associated; a necessary prerequisite to obtaining statistics about this association. Fortunately, the issue may not be a critical one, as when an artifact shares a significant amount of time with two real recognitions it usually has an anomalous length (presumably reflected in the probability of occurrence of its intrinsic characteristics) or an unusual inter-word, start or end delay. The two relationships introduced here are thus seen to be complementary. Real/artifact association statistics are most effective in detecting artifacts which are nearly coincident with real recognitions, and delay considerations (start, inter-word and end) are most effective relative to artifacts generated by the juncture of two words. (It is feasible to consider artifacts in association with real word pairs, rather than with single words. However, more data are required to obtain reliable statistics about these associations, and more data are required to use them in the recognition process. We therefore restrict our attention to the simpler case.)

Formulation of the Word Interaction Problem as One in Statistical Decision Theory

We now develop a mathematical characterization of the problem of determining which of a set of potential recognitions should be considered real, and constructing the final output of the LCSR recognition algorithm. First we formulate the problem as one of finding the best path through a certain acyclic directed graph.

The ordered occurrence of recognitions by the individual word spotting machines induces natural limitations on the order in which the recognized words were spoken. If "five" was recognized over one time interval, and "six" over an entirely disjoint interval, there is only one plausible order in which the subject "five" and the subject "six" were spoken. Furthermore, a given recognition will have only a limited collection of plausible immediate predecessors. One can establish lower and upper limits, Δt_{SM} and Δt_S, on the time interval between recognitions by surveying training data. Let the set of potential recognitions (received from MEX) be $\Pi = \{\pi_1, \ldots \pi_N\}$. If $TR(\pi)$ is the recognition time of the potential recognition π, then we consider π' a potential immediate predecessor of the recognition π when

$$\Delta t_{SM} \leq TR(\pi) - TR(\pi') \leq \Delta t_S.$$

This condition is also used to determine which potential recognitions are potential first words and last words of the utterance, by introducing the pseudo-recognitions called Start and End, with recognition times equal to zero and the end-of-utterance time, respectively. That is, potential recognition π is considered a potential first recognition when

$$\Delta t_{SM} \leq (TR(\pi) - TR(\text{"Start"})) = TR(\pi) \leq \Delta t_S$$

and a potential last recognition when

$$\Delta t_{SM} \leq (TR(\text{End}) - TR(\pi)) = (t_{EOU} - TR(\pi)) \leq \Delta t_S.$$

This condition defines an asymmetric (assuming $\Delta t_{SM} > 0$) relation on a set of nodes defined to be

$$N = \Pi \cup \{\text{Start, End}\}.$$

The relation is true of n and n' where n and n' are nodes, precisely when n is a potential immediate predecessor of n'. Denote the relation PP(n, n').

The set of nodes, N, and the relation PP together are equivalent to a directed graph, illustrated in figure 16. The statistics of inter-word delays are such that for any two potential recognitions one can always eliminate, with perfect confidence, either the possibility that one follows the other, or that the other follows the one. The relation PP is therefore irreflexive and antisymmetric, and Δt_S is such that every potential recognition has at least one potential predecessor, and such that at least one potential recognition is a potential first recognition and at least one is a potential last recognition. We further note that every potential recognition is the potential immediate predecessor of at least one other potential recognition or else is a potential last recognition. Under these conditions the directed graph of nodes is connected in the sense that there is a path from any given node to the Start node. The directed graph is also acyclic.

In terms of the directed graph representing the utterance, the problem at hand is to find the best path (backwards along the arrows) from Start to End. Data to be considered in finding the best path include the intrinsic data, associated with recognition node; start, inter-word and end delay data which is naturally associated with edges, and association data.

Statistical decision theory may be used to rationalize an intuitively appealing solution to the problem just posed. Under this approach, each path through the directed graph of recognitions from Start to End is considered an hypothesis, and the problem is considered one of choosing the hypothesis which best, in some sense, explains the observed data.

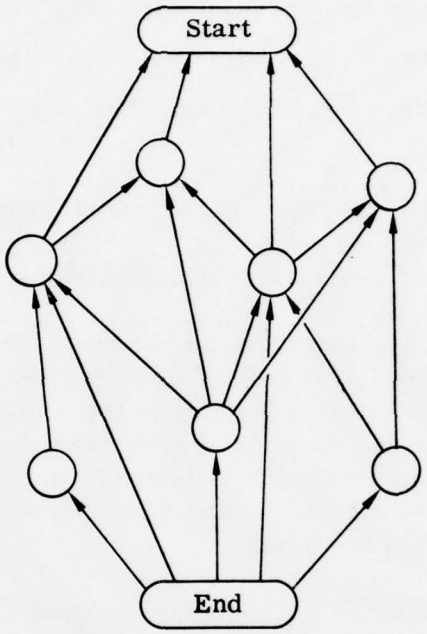


Figure 16. Directed Graph Representing Real Recognitions and Artifacts, and the Potential Immediate Successors Relation. (Circles represent nodes, and there is an edge from node i to node j precisely when PP(j, i)).

The set of hypotheses of interest is the set of all paths from the Start node to the End node. To each hypothesis there then corresponds an explanation or interpretation of the utterance as a sequence of spoken words, $v_1 v_2, \ldots v_k$, where each v_i is the vocabulary item corresponding to the potential recognition found in following the hypothetical path Start, π_{i_1}, $\pi_{i_2}, \ldots \pi_{i_k}$, End, from Start to End. Note that two different hypotheses (paths) may have the same underlying interpretation of the utterance, and that we make no use of this fact.

To apply statistical decision theory to the problem of finding the best hypothesis, and hopefully therefore the best (most frequently correct) interpretation of the utterance, we treat the operation of MEX as a process of

observation, and ultimately argue that we should choose the interpretation corresponding to the hypothesis which has the highest a posteriori probability of being true, in light of the observation. To carry out this program we must first be explicit about what constitutes the observation.

MEX provides the following data about the utterance:

a. The occurrence of potential recognitions $\Pi = \{\pi_1, \ldots \pi_n\}$.

b. Intrinsic characteristics of each potential recognition $I(\pi)$. Specifically, these intrinsic characteristics are the loop letter set and transition letter set counter Q_T and Q_L, and violation category QCAT. But for the machine interaction problem it does not matter what the intrinsic data elements are, but only that we can know a priori conditional probabilities Prob $\{I(\pi) \| \pi$ is Real$\}$ and Prob $\{I(\pi) | \pi$ is an Artifact$\}$.

c. The potential predecessor relation on nodes. That is, MEX tells us (data from which we can discern) which potential recognitions are potential first words, last words, and potential immediate predecessors of other nodes.

d. Among potential first words and last words, the actual time interval between the start of the utterance and the recognition start time, or between the recognition time and the end of the utterance. Also, among pairs of recognitions related as potential immediate predecessors, the overlap or gap (underlap) between the recognition time of the predecessor and the start time of the successor.

e. The absolute start and end time of each potential recognition, from which can be computed and the coincidence time of pairs of recognitions, which is indicative of the likelihood that one recognition is real, and causes another as artifact.

In reality MEX can derive much more data from the utterance than is mentioned here. For example, when violations occur, MEX determines specifically which transition or loop letter set is violated, and even which specific features are violated. The data elements enumerated above are, on the basis of some analysis and some intuition, a selection of elements

believed to be the most indicative of which recognitions are real and which are artifacts. The theory of statistical decision making lets us define what constitutes the observation in any way we wish, and then shows how best to use the data retained to reach a decision. One may ignore as much data or information as one likes, to simplify the data gathering problem and the decision process, and the theory will still tell how to proceed to a decision. Of course the more information discarded beyond the essential amount, the more frequently the decision will be in error. Hopefully, the selection of what to retain and what to discard is an optimal trade-off between recognition accuracy on one hand, and data gathering and computational burdens on the other.

An example of discarding data is evident in items c and d above. The start delay, end delay and overlap/gap can be computed easily from the start and recognition times of any potential recognitions or pair of potential recognitions. But it is intuitively clear that these data are only really very relevant for potential first words, potential last words and potential immediate predecessors, respectively. Other time intervals are ignored. We must now proceed to discard more data about the utterance - some of that mentioned in item e, to make the problem more feasible of quick solution. Unfortunately, it is not as clear in this case that the data discarded are as irrelevant as those mentioned in connection with items c and d.

The tendency for a recognition of one type to occur as artifact when a particular vocabulary item is spoken can give very valuable clues to the correct interpretation of an utterance. Utilizing this phenomenon efficiently requires defining a criterion for deciding when one potential recognition shows sufficient coincidence with another to suggest that if one is real it is causing the other as artifact. The criterion chosen is about the simplest possible; that the coincidence time should exceed a threshold which is machine-type dependent. When it occurs that the coincidence time meets or exceeds the criterion for a potential recognition machine type, that potential

recognition is said to be associated with the coincident recognition. Note that, as the coincidence criterion is machine-type dependent, association defined in this way is not necessarily a symmetric relation.

When association is defined in terms of the coincidence time of potential recognitions, it becomes a binary relation on nodes. The observation provided by MEX might then be construed to include this binary relation (as well as the potential predecessor binary relation). However, doing so complicates the problem considerably, as it is difficult to compute the probability of occurrence of both the association and predecessor relations. Among other things, they are definitely not statistically independent characteristics. The problem may be simplified by disregarding the specific recognitions associated with a given recognition and noting only the machine types of the associated recognitions. We even disregard the number of recognitions of a given type which are associated, considering only whether or not a recognition by a machine of a given type is associated. This reduces the association aspect of an observation to a record of the types of machines with which a given recognition is associated. (i.e., to a function from the set of potential recognitions into the power set of machine types.) As such it is information about each individual potential recognition; not about pairs. Reduced in this way, the association component of the observation of an utterance is simplified and made more concrete. We therefore replace item e given above by:

e'. For each potential recognition, π, $A(\pi)$, which is the set of types of machines associated with π.

The data composing an observation of an utterance are now neatly depicted in an annotated directed graph, as in Figure 17.

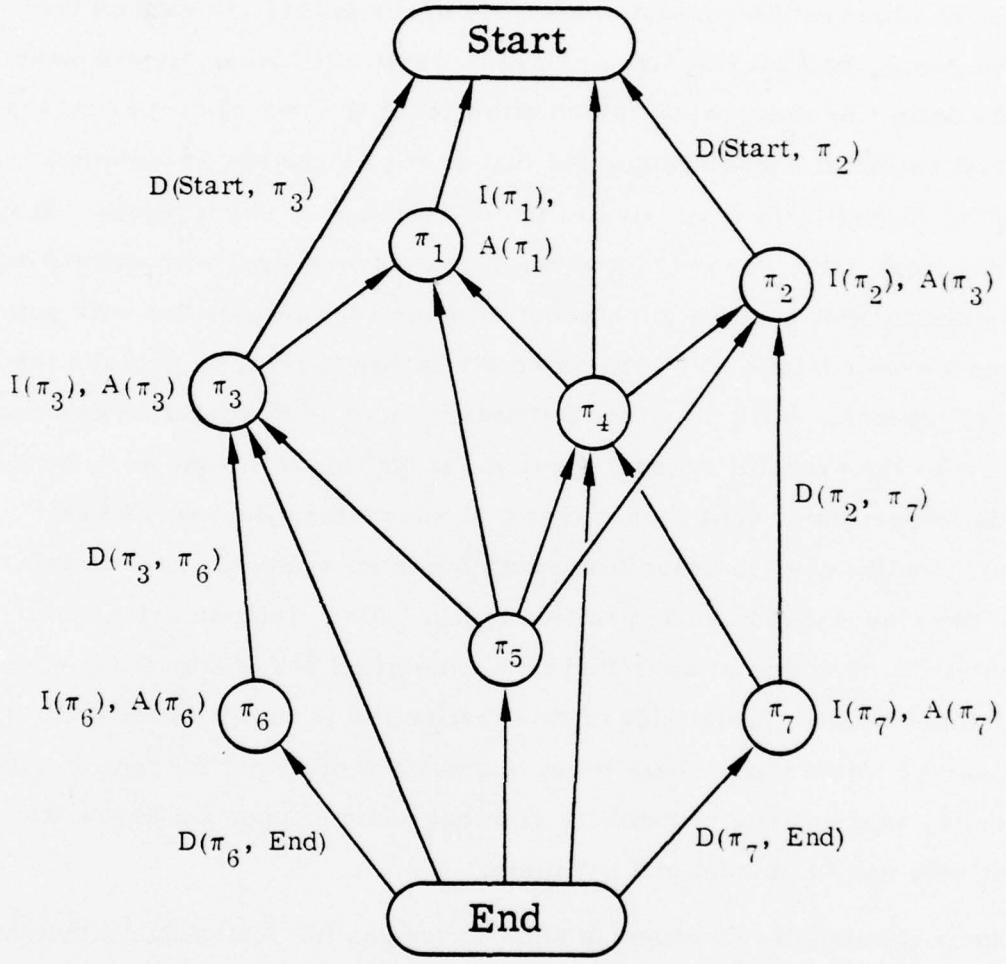


Figure 17. Annotated Directed Graph depicting the data comprising an observation of an utterance. With each potential recognition, π, there is associated $I(\pi)$, the intrinsic data, and $A(\pi)$, the set of associated machine types. With each edge of the graph is associated a delay (overlap or gap time) value, D. Only a few of the data associated with nodes and edges are shown.

The final step in defining what constitutes an observation is crucial to the rigorous derivation of the solution procedure. By definition, we establish that an observation consists precisely of the annotation data on the directed graph, and not the directed graph itself. It may at first appear that this definition disregards information about the fact of occurrence of individual potential recognitions, but that is not so, as the association data entail (some) data indicative of the occurrence of recognitions. It is critically important, however, that we do not have to deal with computing the probability that precisely the set Π of potential recognitions with potential predecessor relation PP will arise given that $\pi_1, \pi_2 \cdots \pi_K$ are the real recognitions. With the proposed definition of an observation one deals instead with the probability that, given a recognition of a real word by a machine of particular type (hence that that vocabulary item was really spoken), it will occur in association with other (presumably artifactual), recognitions by a machine of specified types. Also given an artifactual recognition by machine of specified type, specified associates will be recognized. The critical importance of this distinction is that it is difficult to model and estimate the probability of occurrence of Π and PP from a given hypothesis, whereas the probability of occurrence of specific associations is relatively easy to model and estimate.

The preceding development is summarized in the following mathematical model of the machine interaction problem. There are given:

a. A directed graph G = (N, E) with vertices (called nodes)

$$N = \{\text{Start, End}\} \cup \Pi \quad (\Pi = \{\pi_1, \ldots \pi_n\})$$

and edges $E \subset N \times N$. G is acyclic and connected in the sense that there are paths from every node to the start node.

b. A set of machine types, M

c. A function m: $\Pi \rightarrow M$, where $m(\pi)$ is the machine type of node π.

d. A set K of possible intrinsic characteristics of a node $\pi \in \Pi$.

An observation consists of three functions:

a. $D:E \rightarrow J$, the (integer-valued) delay, overlap or gap associated with an edge of the graph.

b. $I:\Pi \rightarrow K$, where K is the set of all possible intrinsic characteristics (observed about) a node.

c. $A:\Pi \rightarrow P(M)$, where $A(\pi)$ is the set of machine types associated with the node $\pi \epsilon \Pi$, and P denotes the power set.

The set, H, of all hypotheses is the set of all backward paths in G from Start to End.

The Statistical Decision theory approach to the problem at hand entails invoking Bayes' Theorem, which yields, for any hypothesis $h \epsilon H$.

$$\text{Prob}(h \mid \text{Observation}) = \left[\frac{\text{Prob}(h)}{\text{Prob}(\text{Observation})}\right] \text{Prob}(\text{Observation} \mid h)$$

In the absence of any concept of relative cost of various types of possible errors, we opt to accept the hypothesis which has the greatest probability of being true, given the observation. (But note that the resulting treatment is easily modified to accommodate costs assigned to various types of errors. Such costs can be meaningfully assigned in some concrete applications of ASR.)

The a priori probability of a given hypothesis, h, being true, Prob (h), can be estimated as the product of probabilities that the recognition in the hypothetical path are in fact real, times the product of the probabilities that the recognitions not in the hypothetical path are in fact artifacts. These probabilities can be estimated from training data with the same distribution of vocabulary items expected in application of the recognition procedure. (Again, in a concrete SR application, syntactic, semantic or task-related information may be used to estimate different a priori probabilities of various hypotheses.) Under these assumptions, the a posteriori probability of a given hypothesis being true, given the observation, is directly proportional to the product of the hypothesis' a priori probability and the probability that the observation would occur, given that the hypothesis is true. Since the

constant of proportionality is in fact constant over all hypotheses, the problem is properly solved by selecting the hypothesis which maximizes this product, and we now address computing this product.

Let the hypothesis h be Start, π_1, π_2, . . . π_k, End. Assuming independence of the three aspects of an observation, we have

$$\text{Prob (h) Prob (Observation } |\text{h)} = \text{Prob (h) Prob (D } |\text{h) Prob (I } |\text{h) Prob (A } |\text{h)}$$

We now model each of the probabilities on the right. First, let us belatedly assume that it is possible to identify, for any actual utterance, which recognition are real and which are artifacts in such a way that one and only one hypothesis is true. This leads to a natural definition of a real edge of the utterance graph as any edge which is part of the correct hypothesis. Other edges will be called artifact edges, denoted E_A.

Assume that the delays, overlays and gaps are:

a. Independent of one another.

b. Dependent upon only three factors:

F1 the type of machines involved

F2 whether an initial delay, final delay or overlap/gap.

F3 whether the associated edge is real or artifact.

Then

$$\text{Prob (D } |\text{h)} = \text{Prob (D (Start, } \pi_1)|\text{m}(\pi_1) \text{ is real first word mach type)} \times$$

$$\prod_{i=1}^{k-1} \text{Prob (D } (\pi_i, \pi_{i+1})|\text{m}(\pi_i) \text{ and m}(\pi_{i+1}) \text{ are consecutive real mach types)} \times$$

$$\text{Prob (D } (\pi_k, \text{ End})|\text{m}(\pi_k) \text{ is real last word mach type)} \times$$

$$\prod_{(\text{Start}, \pi) \in E_A} \text{Prob (D (Start, } \pi)|\text{m}(\pi) \text{ is not real first word mach type)} \times$$

$$\prod_{(\pi, \pi') \epsilon E_A} \text{Prob } (D(\pi, \pi') \,|\, m(\pi) \text{ and } m(\pi') \text{ are not real consecutive mach types}) \times$$

$$\prod_{(\pi, \text{End}) \epsilon E_A} \text{Prob } (D(\pi, \text{End}) \,|\, m(\pi) \text{ is not real last word mach type})$$

Proceeding similarly for the intrinsic characteristics of each potential recognition, we assume that the intrinsic characteristics observed about a potential recognition are:

a. Independent of one another

b. Dependent only upon two factors:

F1 the type of recognizing machine

F2 whether the recognition is real or an artifact.

Then

$$\text{Prob } (I \,|h) = \prod_{i=1}^{k} \text{Prob } (I(\pi_i) \,|\, m(\pi_i), \text{ real}) \times$$

$$\prod_{i=k+1}^{n} \text{Prob } (I(\pi_i) \,|\, m(\pi_i), \text{ artifact})$$

In computing the probability of occurrence of various associations under given hypotheses, it is necessary to distinguish that recognition which is the first real one (under the hypothesis) and also those other potential recognitions that are potential first words but are artifacts (under the hypothesis). This group of initial real and artifactual recognitions have associations different from the others because some machine types operate only during the first part of an utterance, and other machines only later. Under the hypothesis h, π_1 is the initial real recognition. Let Π_{AI} denote the set of initial artifactual recognitions, and Π_{AN} the remainder of the artifactual recognitions (under h).

We assume that the probability of a given potential recognition being associated with one or more others of a given machine type is:

a. Independent of any other associations.

b. Dependent only upon factors:

F1 the machine types involved

F2 whether the given recognition is real or artifactual.

F3 whether the given recognition is initial or not initial.

Under these assumptions

$$\text{Prob}(A|h) = \prod_{i=1}^{n} \text{Prob}(A(\pi_i)|h)$$

$$= \text{Prob}(A(\pi_1)|\text{real, initial}) \times$$

$$\prod_{i=2}^{k} \text{Prob}(A(\pi_i)|\text{real, non-initial}) \times$$

$$\prod_{\pi \in \Pi_{AI}} \text{Prob}(A(\pi)|\text{artifact, initial}) \times$$

$$\prod_{\pi \in \Pi_{AN}} \text{Prob}(A(\pi)|\text{artifact, non-initial})$$

Let $B(m, m')$ denote the occurrence of association between machines of type m and m' (according to the criterion for m). Also define two other terms:

$$L(m|C_1, C_2) = \prod_{m' \in M} (1 - \text{Prob}(B(m, m')|C_1, C_2))$$

$$\lambda(m, m'|C_1, C_2) = \frac{\text{Prob}(B(m, m')|C_1, C_2)}{1 - \text{Prob}(B(m, m')|C_1, C_2)}$$

where C_1 is the condition real or artifact, and C_2 is the condition initial or final. Then each factor in Prob $(A|h)$ can be expressed as

$$\text{Prob}\,(A(\pi_i)|C_1, C_2) = L(m(\pi_i)|C_1, C_2) \prod_{m' \epsilon A(\pi_i)} \lambda(m(\pi_i), m'|C_1, C_2)$$

Finally, the a priori probability of the hypothesis h can be expressed as

$$\text{Prob}\,(h) = \prod_{i=1}^{k} \text{Prob}\,(m(\pi_i) \text{ is real}) \times \prod_{i=k+1}^{n} \text{Prob}\,(m(\pi_i) \text{ is artifact})$$

Using the probabilistic models just presented, the probability of occurrence of the total observation can be computed under all possible hypotheses. The computation, as presented, would entail the product of many terms; one for each data item in the directed graph annotation. That is, there is a term for each value D on each edge of the graph, and a term for each label A and I on each node of the graph. Each term represents a probability, hence is non-negative. If we assume probabilities equal to zero do not occur (or extend our arithmetic to allow addition of infinity), then each term can be replaced by its negative natural logarithm, becoming a non-negative value. Under this artifice, the negative natural logarithm of the probability of occurrence of the observation under an hypothesis is just the sum of all appropriate terms at nodes and edges of the graph. The problem of hypothesis selection becomes one of selecting the hypothesis which minimizes this large sum.

An Important Transformation — One further transformation of the problem renders it easily solvable by dynamic programming. Each node and edge in the graph has two sets of labels, one to be used if the nodes or edge lies on the hypothesis path, and another value to be used when it does not. The value assigned to any particular hypothesis is then the sum over the entire graph of the non-hypothesis values, plus the sum, for every node and edge on the hypothesis path, of the differences between the hypothesis and

non-hypothesis values. But the sum over the entire graph is a term common to all hypotheses, and so can be ignored. The best hypothesis is therefore the one which has minimum sum over its nodes and edges of the difference of the two types of values. The problem is therefore one of finding the minimum cost path from the Start node to the End node, where the cost of a path is construed to mean the sum of the difference values at each node and edge of the path.

Casting the problem in terms of logarithms of conditions probabilities and showing that the important values to associate with nodes and edges are differences of these logarithms when the probabilities are conditional upon real and artifact shows that what are really important are the likelihood ratios, computed from the observed data. We now make this explicit, by naming and exhibiting the values to assign to each node and each edge of the graph when solving the problem in its least cost form.

With each node, $\pi \epsilon \Pi$ we associate the cost

$$\Delta Q(\pi) = \Delta Q^{ap}(\pi) + \Delta Q^{I}(\pi) + \Delta Q^{A}(\pi)$$

where

$$\Delta Q^{ap}(\pi) = - \ell n \left[\frac{\text{Prob } (m(\pi) \text{ is real})}{\text{Prob } (m(\pi) \text{ is artifact})} \right]$$

$$\Delta Q^{I}(\pi) = - \ell n \left[\frac{\text{Prob } (I(\pi) \mid m(\pi), \text{ real})}{\text{Prob } (I(\pi) \mid m(\pi), \text{ artifact})} \right]$$

and

$$\Delta Q^{A}(\pi) = - \ell n \left[\frac{\text{Prob } (A(\pi) \mid \text{real})}{\text{Prob } (A(\pi) \mid \text{artifact})} \right]$$

$$= - \ell n \left[\frac{L(m(\pi) \mid \text{real, C})}{L(m(\pi) \mid \text{artifact, C})} \right]$$

$$- \sum_{m' \epsilon A(\pi)} \ell n \left[\frac{\lambda(m(\pi), m' \mid \text{real, C})}{\lambda(m(\pi), m' \mid \text{artifact, C})} \right]$$

where C is the condition initial or non-initial, as appropriate to π.

With each edge of the graph, we associate a cost value depending upon the observed delay, overlap or gap. The value is given by the following expressions:

$$\Delta Q^{G}(\text{Start}, \pi) = -\ell n \left[\frac{\text{Prob}\,(D(\text{Start}, \pi) \,|\, m(\pi) \text{ is real first word mach. type})}{\text{Prob}\,(D(\text{Start}, \pi) \,|\, m(\pi) \text{ is not real first word mach. type})} \right]$$

$$\Delta Q^{G}(\pi, \pi') = -\ell n \left[\frac{\text{Prob}\,(D(\pi, \pi') \,|\, m(\pi) \text{ and } m(\pi') \text{ are consecutive, real mach. types})}{\text{Prob}\,(D(\pi, \pi') \,|\, m(\pi) \text{ and } m(\pi') \text{ are not consecutive, real mach. types})} \right]$$

$$\Delta Q^{G}(\pi, \text{End}) = -\ell n \left[\frac{\text{Prob}\,(D(\pi, \text{End}) \,|\, m(\pi) \text{ is real last word mach. type})}{\text{Prob}\,(D(\pi, \text{End}) \,|\, m(\pi) \text{ is not real last word mach. type})} \right]$$

The cost associated with the hypothesis h = *Start*, π_1, π_2, . . . π_k, *End* is then

$$Q(h) = \sum_{i=1}^{k} \Delta Q(\pi) + \Delta Q^{G}(\text{Start}, \pi_1) + \sum_{i=1}^{k-1} \Delta Q^{G}(\pi_i, \pi_{i+1}) + \Delta Q^{G}(\pi_k, \text{End})$$

Minimizing this cost over all hypotheses identifies the hypothesis with greatest a posteriori probability.

Solution of the Problem by Dynamic Programming

The dynamic programming solution hangs on the following facts. Let $P \subset N$ be a set of nodes closed with respect to the operation of finding potential immediate predecessors; i.e., such that if $m \in P$ and m' is a potential immediate predecessor of m, then m' is also in P. Let P be equipped with all the edges it inherits from the original graph on N; thus equipped it becomes a complete subgraph of the original graph, and is connected and acyclic. For any node m in P and any backward path Start $m_1 m_2 \text{--} m_k m$, define a quality measure

$$q = \sum_{i=1}^{k} Q(m_i) + \Delta Q^G(\text{Start}, m_1) + \sum_{i=1}^{k-1} \Delta Q^G(m_i, m_{i+1}) + \Delta Q^G(m_k, m).$$

Let h(m) be a backwards path from m to S (within P) for which q attains a minimal value, and associate that minimal value with the node m, denoted by q*(m). Notice that this definition of h(m) and q*(m) does not depend upon the particular set of nodes P, as long as it is predecessor complete and contains m.

The original set of nodes, N, has the properties required of P, and if we define

$$Q(\text{End}) = 0$$

then h(End) is the solution to the entire problem, as the objective function Q for this hypothesis is

$$Q(h) = q^*(\text{End}),$$

which shows that h (End) is the hypothesis which minimizes Q.

The dynamic programming solution then follows immediately from the recursive properties of the quality measure, q, which assures that q can be minimized by processing each node once, rather than each hypothesis once. To make the method explicit, let node m have potential immediate predecessors $m_1, m_2, \ldots m_p$, and consider h(m), the optimal backward path from Start to m. As this path must pass through one of the nodes $m_1, \ldots m_p$, it is $h(m_i)\ m_i$ for some i. The quality measure of this path is

$$q = q^*(m_i) + \Delta Q^G(m_i, m) + \Delta Q(m).$$

The last term is common to all paths leading to m, so the quality measure of the optimal path leading to m is

$$q^*(m) = \min_i \left(q^*(m_i) + \Delta Q^G(m_i, m)\right) + \Delta Q(m)$$

The problem is therefore efficiently solved by computing the value q*(m) for each node when its potential predecessors are known, and storing a pointer with each node leading to its optimal predecessor. The optimal predecessor of the new node is found according to the equation just given, from the q* values of its immediate predecessors only, and does not require looking further up the graph of nodes. The final solution is obtained when the optimal predecessor of End is found, by retracing the upward pointers. The process is started by letting h (Start) = Start and q* (Start) = 0.

Real Time Considerations — When notification of a potential recognition is received from MEX, the potential predecessors of that recognition will already have been received. As a result, the quality value, q*, and best immediate predecessor of a node can be computed in terms of the nodes already so processed. The solution to the whole recognition problem can therefore be constructed as potential recognitions are received. In fact, it is not always necessary to wait until the end of the utterance to run up the optimal predecessor pointers and find words to emit as recognized. As soon as all active upward paths have a node in common, that node can be emitted as part of the optimal hypothesis; i. e., a recognized word. The nodes which have been emitted as final recognitions, also (except for the most recent) can be deleted from memory storage. Because of this, two aspects of the machine interaction algorithm are intimately related: the ability to emit a word as recognized before the end of the utterance, and the ability to treat an utterance of arbitrary length with a finite amount of storage. The algorithm described below has both these desirable properties.

Not surprisingly, the early-emission and storage saving capabilities complicate the machine interaction algorithm somewhat. Another source of complication is the need to delay computation of the optimal immediate predecessor of an incoming node until it can safely be assumed that all recognitions with which it might be associated have been received, as these must be known before the Q^A term is computable.

Description of the MINT Algorithm

Processing of potential recognitions in the MINT algorithm is most easily described in terms of nodes, which may be either potential recognitions or artificial recognitions labeled "Start" or "End." The algorithm is initialized with a single node, the Start node, present, and a new node is created each time MEX sends notification to MINT that a new potential recognition has been detected. MEX also informs MINT when the end of the utterance is detected by the preprocessor device handler, and MINT creates the End node at that time. Processing of nodes is, for the most part, identical for the Start and End nodes and for nodes representing potential recognitions.

In the implementation of this algorithm each node is linked to the node received next (downlinks), and each node is also eventually linked to its optimal predecessor (uplinks). As indicated in the derivation of the algorithm, each node (except Start) has a unique predecessor, but may have an arbitrary number of successors; i.e., nodes for which it is the optimal predecessor. The number of nodes for which a given node is the optimal predecessor is called the S-count (successor count) of the node.

Nodes are considered to lie in one of three groups. Upon receipt, a node is made a member of Group III. It is eventually moved into Group II and later either is deleted or moves into Group I where it remains until deleted. The oldest node still remaining is denoted Top. Defining characteristics of the three groups are:

Group I — Nodes whose successors are known, but which cannot be deleted because the real successor of Top is not yet known.

Group II — Nodes whose associated recognitions have all been identified and for which the q* value and optimal predecessor have been calculated. All potential predecessors of the oldest node in Group III are in this group. Successors of these nodes may not be known.

Group III — Nodes received recently enough that another associated potential recognition may yet be received, hence these are nodes for which q* cannot yet be calculated.

Initially Groups I and III are empty, and the Start node, with q* and S-count equal to zero, is the only member of Group II. As it is the oldest node remaining, it is Top.

Processing is facilitated by maintaining a variable called the Number of Uplinks. This is the number of nodes in Group II for which some node in Group I is the optimal predecessor. As Group I is initially empty, the Number of Uplinks is initially zero, but it need not be initialized, as it is set during processing.

The data which must be stored for each node (in addition to bookkeeping) are the machine and vocabulary item (which can also be given special values for identifying the Start and End nodes), a variable q for accumulating q*, the S-count, and an uplink to its optimal predecessor. Temporary storage must also be provided for accumulating the set of machine types with which a node is associated, while the node is in Group III.

Flow Charts – Figures 18 through 21 are flow charts which describe the MINT algorithm at a rather high level. Processing is initiated by receipt of notification from MEX of a potential recognition. Double-ended boxes on these flowcharts indicate procedures described on succeeding pages.

The action shown on Figure 19 is taken on nodes which have been in Group III for enough time to assure that all recognitions with which it may be associated have been received. Therefore ΔQ^A, the quality contribution which reflects the nodes associations can be computed, completing the computation of all the components of q which depend upon the node alone. As all of its potential predecessors are also known, its optimal predecessor can be found and the computation of q* completed.

The action shown on Figure 20 is taken on a node which has been in Group II so long that all of its successors are known. If there are none, it and possibly some of its optimal predecessors can be discarded. If it has

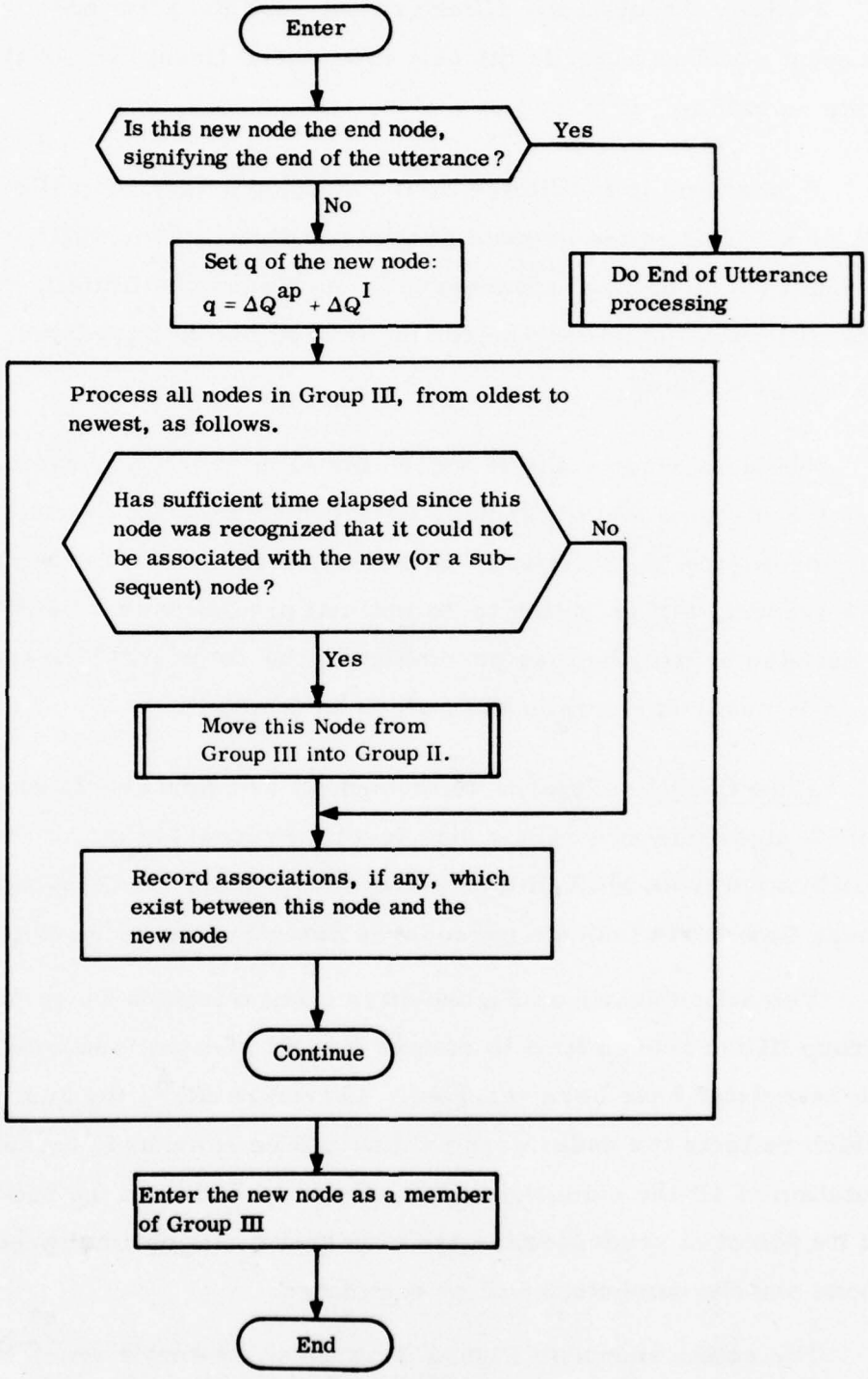


Figure 18. Receive a New Node

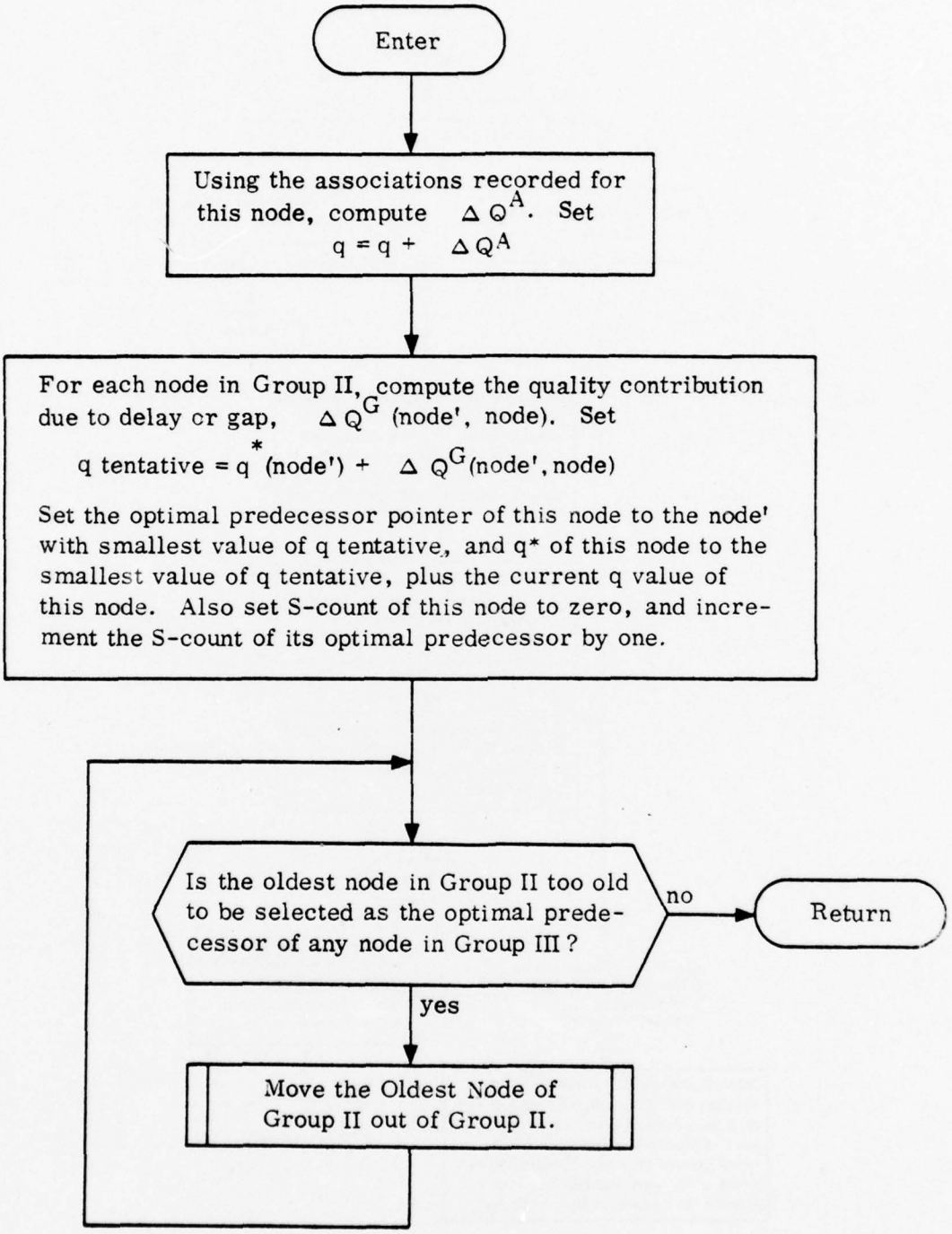


Figure 19. Move This Node from Group III into Group II

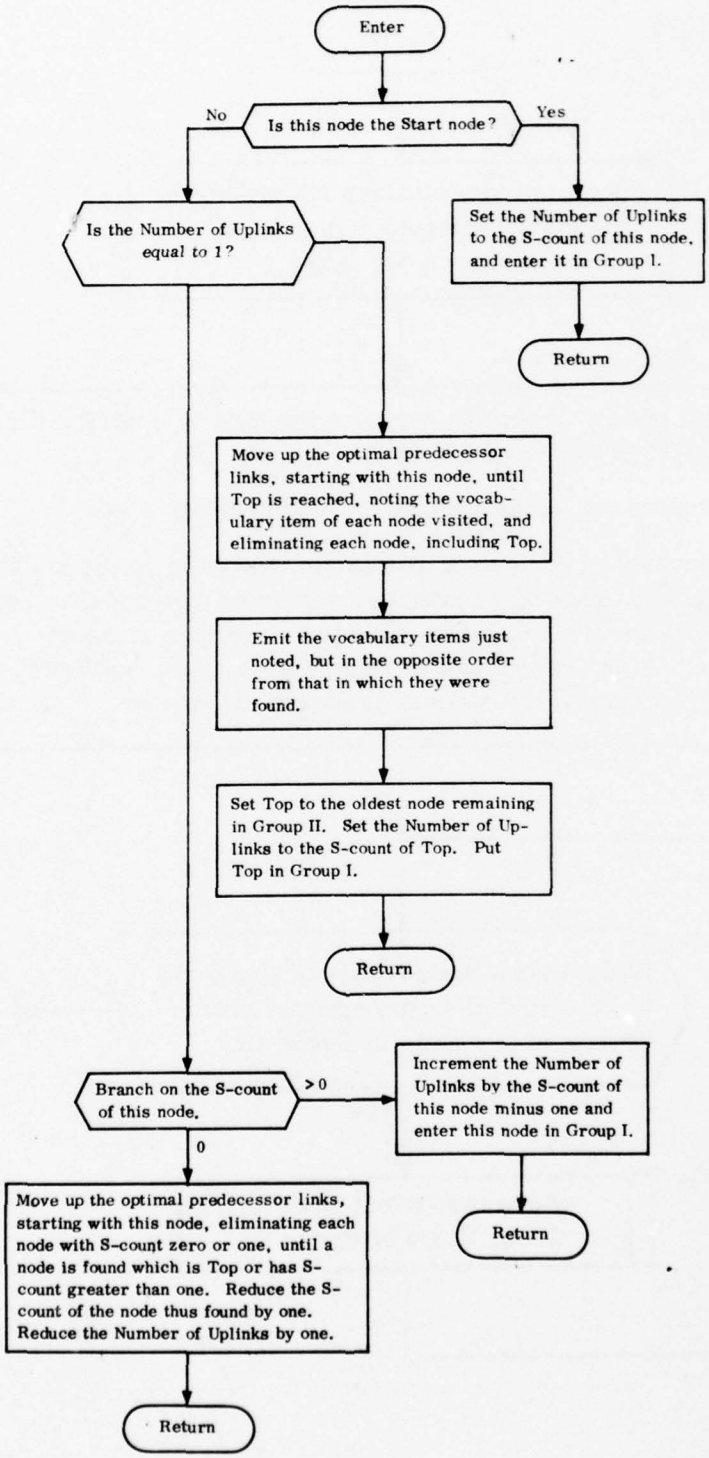


Figure 20. Move the Oldest Node of Group II Out of Group II

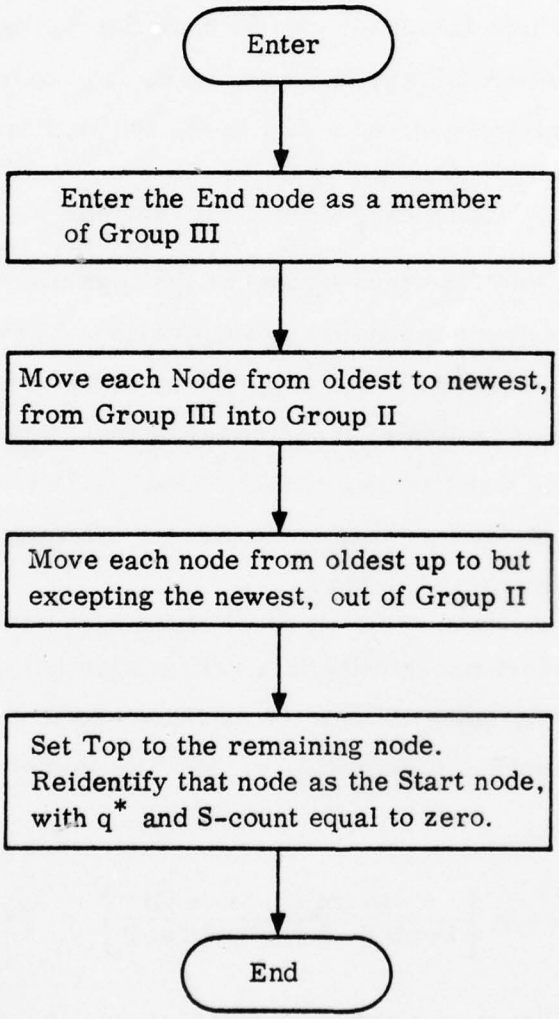


Figure 21. Do End of Utterance Processing

one or more successors it may enable output of a recognition, or it may have to be moved into Group I, awaiting further information.

As shown in Figure 21, when the end of utterance notification is received from MEX, there are usually several nodes in Groups I, II and III. The processing is completed by using procedures already described to move all nodes from Group III into Group II, and then all but the last node out of Group II. To reinitialize for another utterance, it is only necessary to re-identify the one remaining node (the End node) as the Start node.

Computing Cost Contributions

Each of the cost contributions (ΔQ's) is the logarithm of a likelihood ratio, i.e., the ratio of two conditional probabilities. These ΔQ values are computed using the observed values supplied by MEX and statistical models of the conditional distributions of those observations, given that the word is real and given that the word is an artifact. The statistical models for each observable, and the method used to estimate the ΔQ values and the statistical parameters are discussed below.

$\underline{\Delta Q^{ap}}$ – The a priori probability of a recognition being real or artifact is assumed to depend only upon the type of machine which caused the potential recognition. For potential recognition π, ΔQ^{ap} is therefore a function only of $m(\pi)$.

$$\Delta Q^{ap}(\pi) = -\ln\left[\frac{\text{Prob}\,(m(\pi)\text{ is real})}{\text{Prob}\,(m(\pi)\text{ is artifact})}\right] = f(m(\pi))$$

The a priori probabilities of a recognition being real or artifact were estimated for each machine type to be proportional to the number of real and artifact recognitions produced in Interim Test Data, which has an approximately uniform distribution of vocabulary items.

$\underline{\Delta Q^I}$ – The intrinsic qualities of each recognition which are observed by MEX are the T and L-Counter quality functions, Q_T and Q_L, and the quality category. (These are defined and described in connection with the MEX algorithm, Section III.) These three observables are assumed to be independent, so that

$$\Delta Q^I(\pi) = \Delta Q^T(\pi) + \Delta Q^L(\pi) + \Delta Q^V(\pi).$$

where

$$\Delta Q^T(\pi) = -\ell n \left[\frac{\text{Prob}\,(Q_T(\pi) = Q_T \pm \Delta \mid \pi \text{ is real})}{\text{Prob}\,(Q_T(\pi) = Q_T \pm \Delta \mid \pi \text{ is artifact})} \right] = f(Q_T(\pi))$$

$$\Delta Q^L(\pi) = -\ell n \left[\frac{\text{Prob}\,(Q_L(\pi) = Q_L \pm \Delta \mid \pi \text{ is real})}{\text{Prob}\,(Q_L(\pi) = Q_L \pm \Delta \mid \pi \text{ is artifact})} \right] = g(Q_L(\pi))$$

and

$$\Delta Q^V = -\ell n \left[\frac{\text{Prob (quality category} \mid \pi \text{ is real)}}{\text{Prob (quality category} \mid \pi \text{ is artifact)}} \right]$$

$$= h \text{ (quality category)}.$$

The function f, used to estimate the log likelihood ratio as a function of Q_T, was first determined by obtaining graphs of the cumulative distribution of Q_T for real and for artifact recognitions, measuring slopes of these distributions at corresponding values of Q_T, and plotting the negative natural logarithm of the ratio of slopes. An upper bound on Q_T was then established, beyond which the ΔQ^T is set to a value corresponding to a likelihood ratio of 10^{-4}. A parabola was then fit to the computed log likelihood values, for values of Q_T less than the upper bound. Coefficients α_i were thus obtained for estimating ΔQ^T as

$$\Delta Q^T(\pi) = f(Q_T(\pi)) = \alpha_0 + Q_T(\pi)(\alpha_1 + Q_T(\pi)\alpha_2)$$

As this procedure was carried out for each machine type, the coefficients α_i are functions of the machine type, $m(\pi)$.

The function, g, used to estimate the log likelihood ratio ΔQ as a function of Q_L, is based on the assumption that Q_L is a mixed distribution, with finite probability of being zero and an exponential distribution of positive values, as illustrated in Figure 22.

Under this assumption (which is based on observation of the distribution actually exhibited by Q_L), ΔQ^L is given by

$$\Delta Q^L = g(Q_L) = \begin{cases} -\ell n\left(p_0{}^R / p_0{}^A\right) \text{ if } Q_L = 0 \\ -\ell n\left[\dfrac{\lambda^R\left(1 - p_0{}^R\right)}{\lambda^A\left(1 - p_0{}^A\right)}\right] + (\lambda^R - \lambda^A) Q_L \text{ if } Q_L > 0 \end{cases}$$

where

$p_0{}^R$, $p_0{}^A$ are the probabilities that $Q_L = 0$ for real and artifactual recognition, respectively

λ^R, λ^A are the parameters of the exponential distribution of positive Q_L values for real and artifactual recognitions.

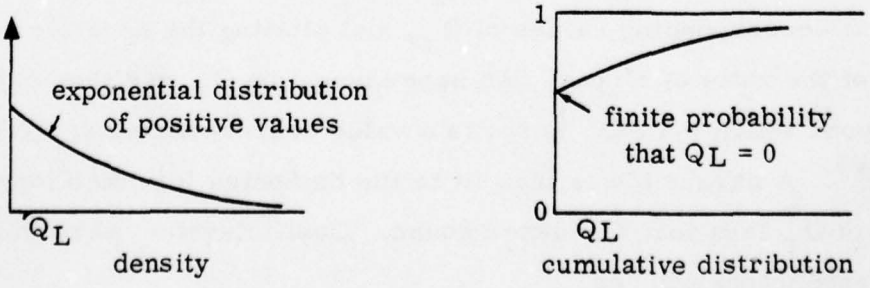


Figure 22. The Distribution Assumed for Q_L

The parameters p_0 and λ were estimated from distributions of Q_L for each machine type, for real and artifactual recognitions in Interim Test data. These were then converted using the above equations to give parameters β_i such that

$$g(Q_L) = \begin{cases} \beta_0 \text{ if } Q_L = 0 \\ \beta_1 + \beta_2 Q_L \text{ if } Q_L > 0 \end{cases}$$

The coefficients β_i are functions of the machine type, $m(\pi)$.

The quality category of a potential recognition indicate the type of transition and loop letter set violation which it exhibited. The probability of occurrence of the various quality categories, and the number of their occurrences in Interim Test data was estimated without regard to machine type. The function h, used to estimate ΔQ^V, is therefore a function of quality category only.

$\underline{\Delta Q^A}$ – As described in the discussion of the concept of association, $A(\pi)$ is the set of machine types with which the potential recognition π is observed to be associated. The probability of occurrence of that set of associated machine types can be computed from a collection of individual probabilities Prob $(A(m, m') \mid C, C')$, where

$A(m, m')$ signifies that a potential recognition by machine type m is associated with a potential recognition by machine type m'

C is the condition initial or non-initial

C' is the condition real or artifact.

These probabilities (for every m, m', C and C') were estimated from the frequency of occurrence of associations observed in Interim Test data.

Several combinations of machine types under some conditions were observed to have probabilities near one or zero of being associated. As the sample size was small in several cases, protection was included to prevent overemphasis on inference from insufficient data. This protection was introduced by using the upper or lower symmetric 50 percent confidence limit, rather than the maximum likelihood estimator for the conditional probabilities. If the frequency ratio for the condition 'real' was greater than the frequency ratio for the condition 'artifact', the lower confidence limit was used for the former and the upper confidence limit was used for latter; that is, those confidence limits were chosen which minimized the difference between estimates of the conditional probabilities for the real and artifactual cases. If the 50 percent confidence limits overlapped, the likelihood ratio was estimated as one.

Before the probabilities of association could be estimated as described above, it was necessary to establish the criterion by which it would be determined that one potential recognition is associated with another; that is, the amount of coincident (overlap) between recognitions by machine types m and m' in order to assert that the former is associated with the latter. (Recall that association is not a symmetric relation.) To establish this criterion, the frequency of associations were observed on Interim Test data for a spectrum of overlap criteria varying from 10 percent to 90 percent of the average duration of recognitions by each machine type. For each overlap criterion, the probabilities $p(A(m, m') \mid C, C')$ were computed as described above, and a Figure of Merit was also computed for each machine type m, and each overlap criterion. The Figure of Merit is an estimate of the average value of ΔQ^A, in the right direction (negative for real recognitions and positive for artifactual recognitions), estimated from Interim Test data. It is given by the following expression.

$$\text{FOM}(m) = \left(\frac{1}{N_A + N_R}\right) \sum_{\substack{\text{Machine} \\ \text{Types}}} \Big\{ [N_A \text{ Prob } (A(m, m') \mid C, \text{artifact})$$

$$- N_R \text{ Prob } (A(m, m') \mid C, \text{real})]\, \Delta Q^{+}$$

$$+ [N_A \, (1 - \text{Prob } (A(m, m') \mid C, \text{artifact}))$$

$$- N_R \, (1 - \text{Prob } (A(m, m') \mid C, \text{real}))]\, \Delta Q^{-} \Big\}$$

where N_R and N_A are the number of real and artifactual recognitions of machine type m, and

$$\Delta Q^{+} = -\ell n \left[\frac{\text{Prob } (A(m, m') \mid C, \text{real})}{\text{Prob } (A(m, m') \mid C, \text{artifact})} \right]$$

$$\Delta Q^{-} = -\ell n \left[\frac{1 - \text{Prob } (A(m, m') \mid C, \text{real})}{1 - \text{Prob } (A(m, m') \mid C, \text{artifact})} \right]$$

The sum in the expression for FOM includes all machine types (both initial and non-initial) if m is an initial type of machine, and just non-initial machine types otherwise. The condition C is initial if m is an initial machine type, and non-initial otherwise.

The overlap criteria for deciding association were then set for each machine type to that value (within the limits observed) which gave maximum FOM.

$\underline{\Delta Q^{G}\ (\text{Start}, \pi)}$ – The distribution of delay time between the beginning of the utterance and the start of the first recognition was observed to be adequately represented by a finite probability of being zero, and an exponentially decreasing probability of being two counts or more. (Delays of one cannot

occur as letters with count of one are eliminated by the preprocessor device handler.) Then ΔQ^G for initial delays is computed from

$$\Delta Q^G(\text{Start}, \pi) = \begin{cases} -\ell n \left(p_0^R / p_0^A \right) & \text{if delay} = 0 \\ -\ell n \left[\dfrac{\lambda^R \left(1 - p_0^R\right)}{\lambda^A \left(1 - p_0^A\right)} \right] + (\text{delay} - 2)(\lambda^R - \lambda^A) & \text{if delay} > 0 \end{cases}$$

where

p_0^R, p_0^A are the probabilities that the start delay will be zero for real and artifactual recognitions, respectively,

λ^R, λ^A are the parameters of the real and artifactual exponential distributions of delays greater than two.

The p_0 and λ values were estimated from start delays observed in Interim Test data, for real and artifactual recognitions. The λ values were estimated as the reciprocal of the mean of non-zero delays minus two. These parameters were computed for each machine type.

$\underline{\Delta Q^G(m, m')}$ – Inter-word gap distributions, between contiguous real recognitions and between other recognitions and their potential predecessors, are assumed to be distributed uniformly over an interval and exponentially outside of that interval. The number of parameters required to describe a member of this class of distributions is reduced to two by assuming that it is symmetric and that the uniform portion of the distribution contains one half of its mass. The distribution is then uniquely determined by its mean and its standard deviation. The uniform portion of this distribution spans ± .6124 standard deviations from the mean, as illustrated in Figure 23. This half-width of the uniform portion is denoted d.

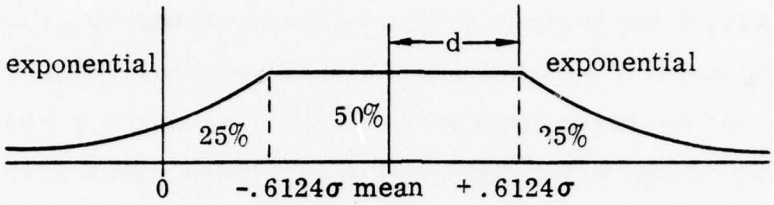


Figure 23. The Distribution Assumed for Inter-Word Gaps

Assuming this distribution, ΔQ^G can be computed as follows.

$$\Delta Q^G(m, m') = -\ell n\,(d^A/d^R) + \Delta Q^{GR} + \Delta Q^{GA}$$

where

$$\Delta Q^{GX} = \begin{cases} 0 \text{ if } |g - \mu^X| \le d^X \\ \dfrac{|g - \mu^X|}{d^X} - 1 \text{ if } |g - \mu^X| > d^X \end{cases} \qquad (X = R \text{ or } A)$$

g is the observed gap

μ^R, μ^A are the means of the contiguous real and other gaps, respectively, for machine types m and m'

d^R, d^A are the half-width parameters of the contiguous real and other gap distributions, respectively.

The parameters d and μ were computed for every machine type pair, for contiguous real recognitions and for other potential neighbors using Interim Test data. The parameter d, for contiguous real recognitions (proportional to the standard deviation of the distribution) was particularly prone to small sample size effects, as the Interim Test Data contained a maximum of six examples of each pair of contiguous real recognitions. To avoid excess emphasis on these limited data, the d values were first computed for all

machine type pairs, and a lower bound was established for each individual machine pair case, lying between the quartile and median value of this distribution varying linearly with the number of cases observed. This limit was defined so that the lower limit was the quartile value if six cases were observed, and was equal to the median if two or fewer cases were observed. After this lower limit was imposed on the d value for each machine pair, it was increased another 50 percent to further deemphasize this information source.

ΔQ^G (m, End) – Real end delay values (the time between recognizing the real last word and the end of the utterance) were assumed to be distributed according to the same symmetric, truncated bi-exponential distribution assumed for interword gaps, but limited to positive values. End delays for potential last words other than the last real recognition were assumed to be uniformly distributed over the range of interest. Under these assumptions ΔQ^G for end delays can be computed from the expression

$$\Delta Q^G(m, End) = -\ell n\,(d^R/u) + \begin{cases} 0 \text{ if } |delay - \mu^R| \le d^R \\ \dfrac{|delay - \mu^R|}{d^R} - 1 \text{ if } |delay - \mu^R| > d^R \end{cases}$$

where

d^R is the half-width of the uniform portion of the real last recognition distribution,

u is a factor incorporating the uniform density of potential last words other than the real one, and a normalizing factor reflecting the truncation of the real end delay distribution,

μ^R is the center of symmetry of the real end delay distribution.

SECTION V

RESULTS, CONCLUSIONS AND RECOMMENDATIONS

In this Section are presented the results obtained in exercising the continuous speech recognition system, LISTEN, on two voices, conclusions which can be drawn from the results of these preliminary tests of the adopted approach to LCSR, and recommendations with respect to continued investigation of this approach.

Results

Real-Time Operation – LISTEN was programmed in Fortran for a Data General Eclipse minicomputer. The MEX and MINT algorithms were implemented with maximum use of linked-list data structures, which are the most efficient form of storage organization for those algorithms in a serial computing environment. Some special coding conventions were used to obtain efficient code generation which efficiently used the stack processing power of the Eclipse, but the coding is primarily conventional.

LISTEN operates in real time, in the sense that the words recognized are emitted by LISTEN within a barely perceptible delay after the speaker stops speaking. This remains true for input utterances of one to fifteen words. (Longer input utterances were not tested.) When long utterances are spoken (over five words) the output comes in staccato bursts during the utterance, usually with two or more words appearing in the last burst at the end of speaking.

When reprogrammed for the slower Data General Nova 800, LISTEN became noticeably slower, requiring as long as ten seconds to recognize a four word utterance. This dramatic reduction in speed is due to the slower

machine cycle, the fact that stack processing is simulated on the Nova 800 but done in hardware in the Eclipse, and because index checking (a compile-time option) was incorporated in the Nova 800 version, to aid in debugging.

Recognition Accuracy – LISTEN was exercised on both Interim Test data and Test data, for speakers MWG and BRO. Both sets of data contained approximately 1050 words in 330 utterances for each speaker. Sixty percent of the utterances in each set of data contained four words, twenty percent three words and the remainder were the vocabulary items spoken in isolation. The utterances were carefully selected so that all contiguous pairs of vocabulary items occurred an equal number of times. The recognition accuracy obtained, computed on an utterance basis, (including insertions, substitutions and deletions as errors) is given below.

	Voice Data Used	
Speaker	Interim Test	Test
MWG	94%	86%
BRO	71%	38%

Recognition Accuracy Observed – (Utterance Basis)

On a word basis, the recognition accuracy on MWG's Test data was 95.3 percent correct. The confusion matrix, showing the vocabulary item dependence of the accuracy, is shown in Figure 24.

Analysis of Errors – A very limited amount of error analysis was performed on the test results, due to the limited time available. A few results of that analysis are presented in the following paragraphs.

The 49-word recognition errors observed on MWG's test data were analyzed to determine which were due to failure of the word-spotting algorithm, MEX, to provide a potential recognition for MINT to select, and which were

WORD ACCURACY

UNDERSTOOD

SPOKEN	0	1	2	3	4	5	6	7	8	9	.	None (Deletions)
0	91		2	2			1					
1		90			3	1						2
2			83	10			1					2
3			2	92	1							
4		1			89						4	1
5						89	1			6		
6							95					
7								96				
8			1						95			
9										96		
.					1	2			.		85	1
None (Insertions)		2			1				1			

Figure 24. Confusion Matrix – 1049 Words (MWG Test Data)
49 Word Mistakes, Including Insertions, Deletions, and Substitutions
95.3% Correct

due to other causes. This form of failure can only arise when the word spoken fails to have several required sounds, or exhibits entirely unexpected sounds several times; i.e. when there are excessive transition or loop letter set violations. The occurrence of structural violations of this kind are not apparent from the confusion matrix, as MINT will often select some other artifactual potential recognition to fill the time gap left by the missing

word, making the error appear to be a substitution. The result of this analysis is shown below.

Structural Violation	10
Substitution	33
Insertion	4
Deletion	2
Total	49

Observed Frequency of Error Types (MWG Test Data)

This analysis shows that approximately 1 percent of the Test sample words exhibited structural peculiarities that were strange enough to prevent potential recognition by MEX. This is approximately three times the rate observed on Interim Test data.

All four insertion errors in MWG's Test data were due to multiple recognitions of the correct word, e.g., "1655" was misrecognized as "11655". This error is made because MINT is presented with two (or more) recognitions which individually are of high quality in terms of observed violations and T and L Counter histories. Both are actually real recognitions, but MINT must find one of them to be an artifactual recognition to avoid insertion. The algorithm could be modified to prevent these double responses quite easily.

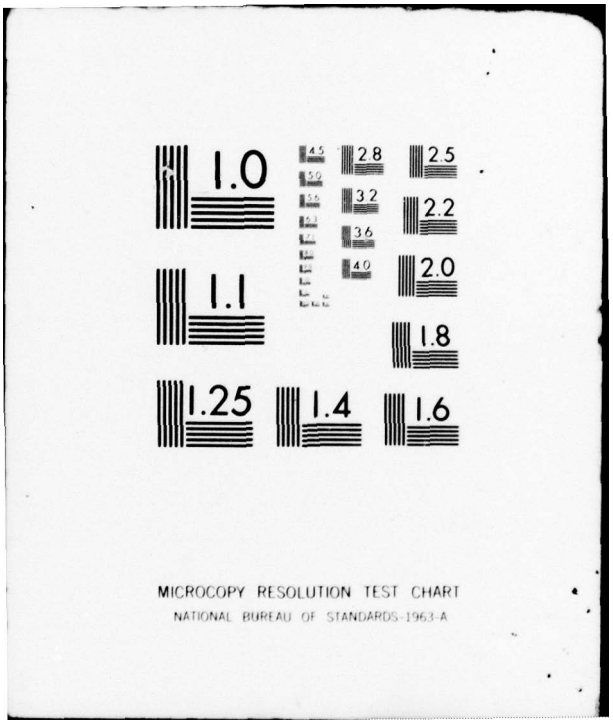
Recognition errors are concentrated in the first word of an utterance, and to a lesser degree in the last word. Substitution errors, rates in multiword utterances, for example, were 6.7 percent in the first word, 1.8 percent in internal words, and 2.3 percent in the last word. This concentration of errors in the first and last words of an utterance causes the utterance error rate to increase more slowly than might be expected as a function of the number of words in the utterance. When one uses the overall word recognition

AD-A056 231 LOGICON INC SAN DIEGO CALIF F/G 9/2
LISTEN: A SYSTEM FOR RECOGNIZING CONNECTED SPEECH OVER SMALL, F--ETC(U)
APR 78 J E PORTER N61339-77-C-0096
UNCLASSIFIED NAVTRAEQUIPC-77-C-0096-1 NL

2 OF 2
AD
A056231



END
DATE
FILMED
8 -78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

rate of 95.3 percent, together with the assumption that errors are independent of the number and location of words in the utterance, the observed recognition rate for isolated words appears low, and for four word utterances appears high, as the table below shows.

Words in Utterance	Observed Recognition Accuracy	Accuracy Predicted by Simple Model
1	93.9	95.3
3	83.1	86.1
4	84.3	82.6

Comparison of Observed Recognition Accuracy with that Predicted Assuming Errors are Independent of the Number and Position of Words in the Utterance

Although time has not allowed careful live testing of LISTEN, it is subjectively apparent that long utterances can be recognized quite well by careful enunciation. Credit card and social security numbers are often recognized without error.

The low accuracy reported for BRO's voice is not well understood at this time, as the errors have not yet been thoroughly analyzed. However, one source of errors was detected, which partially explains the precipitous drop in recognition accuracy between Interim Test data and Test data for BRO. This error source was apparent from the fact that neither the word "point" nor the word "eight" was recognized correctly once on the test data. Upon examining the "eight" errors, it was found that the MEX algorithm did not provide a potential recognition for consideration by MINT, for any of the 96 voicings of "eight" in Test data. The cause of this anomaly was traced to a peculiar change in behavior of the VIP-100 features 20 and 21 relative to features 29 and 30. In every example (192 cases) of Training and Interim Test data, features 29 and 30 ceased to occur before 20 or 21 ceased, and before the end of the word. In Training data this led to

the formation of two transition letter sets for the end of "eight" which required the presence and then the absence of features 29 or 30, while feature 20 or 21 is present. In Test data, and in all voice data recorded by BRO since Interim Test data, these structural features are reversed. In every example of the later voicings of "eight", feature 29 or 30 continues uninterrupted into the following word, leading to excessive transition set violations. This profound, sudden and apparently irreversible change in the character of the VIP-100 output for BRO suggests that a significant change has occurred in the recording environment used by BRO, in BRO's voice or in the VIP-100 used by BRO. The last is a possibility because all of BRO's data were taken using a VIP-100 at NAVTRAEQUIPCEN, while all of MWG's data were taken from a different VIP-100 in San Diego.

Conclusions

Unfortunately, the minimal amount of analysis and testing of the LISTEN system performed to date prevent an unequivocal assessment of the potential of the underlying approach to LCSR. Some firm conclusions can, however, be reached.

It has been demonstrated that the VIP-100 speech preprocessor output, for several individuals, exhibits a rich sequential information structure and that this structure can be extracted automatically. It has also been demonstrated that a recognition algorithm based in the automatically extracted information structure can be implemented to perform recognition in real time. Further conclusions must await further examinations of existing data and additional tests.

Recommendations

The ambiguity of the preliminary test results obtained with LISTEN leaves no alternative but to recommend completing the investigation of its performance. The large discrepancy in performance for MWG and BRO

is particularly important because the 86 percent recognition accuracy obtained on MWG's voice is adequate for some training applications, whereas the 38 percent accuracy obtained for BRO's voice is inadequate for any practical system. Below is a list of questions raised by these preliminary results.

a. A limited number of voice characteristics are derived from the Interim Test data, and these characteristics, such as inter-word durations, were expected to be quite stable over large samples of words. Why, then, are there about two-and-one-half times as many errors over Official Test data as opposed to Interim Test data for MWG?

b. Recognition errors due to structural violations, i.e., due to words which do not have expected sound classes in expected order, and which deviate from these expectations in extreme ways, comprised 0.3 percent of cases (word basis) in MWG's Interim Test data, but 1 percent in his Official Test data. Does this indicate that failure of words to exhibit the needed structure is highly variable and unpredictable?

c. The word spotting portion of the LISTEN algorithm incorporates measures designed to accommodate expected structural violations, based on an analysis of those violations as they occur in MWG's voice. A preliminary analysis indicated a remarkable similarity to MWG's voice in the rate of occurrence of each type of violation, suggesting that the method used to accommodate structural violations should be applicable to speakers other than MWG. Was this preliminary analysis accurate, and if so, what is the source of the large number of failures in BRO's Interim Test Data?

d. All voice data taken from BRO after the Training and Interim Test data show changes in behavior of certain VIP-100 features which cause fatal structural violations in every example of "eight" and "point" in BRO's Official Test and later data. Are these changes due to typical changes in a person's voice over time or are they due to some atypical or even extraneous cause, such as a permanent change in BRO's articulatory apparatus or accidental changes in recording conditions or equipment?

e. Studying the performance of LISTEN on speakers other than MWG in San Diego can help to determine if the large difference in performance for the two speakers indicate that large variability is to be expected across speakers, or that some disrupting influence is affecting the BRO experiments at NAVTRAEQUIPCEN. It is therefore pertinent to determine: how does LISTEN perform for additional speakers in San Diego?

f. A taxonomy of recognition errors has been developed for describing observed errors, and a procedure has been developed for performing the classification. This taxonomy relates errors to the most suspicious information source used in arriving at the erroneous conclusion. Across several speakers, how similar is the rate of occurrence of each type of recognition error?

These questions indicate the importance of a detailed analysis of observed recognition errors. Errors due to structural violations are particularly important, because they can only be reduced, if at all, by changing the word-spotting portion of the algorithm, which requires extensive retesting and may impact the real-time performance of the algorithm. Other sources of error impact primarily the machine interaction portion of the algorithm (which is easier to modify and test, as it is simpler than the word-spotting portion), and the voice data requirements of the algorithm.

The natural course of continuing investigation leading to evaluation and, if possible, improvement of LISTEN is to analyze the observed recognition errors, and relate them to specific aspects of the algorithm. Each such error analysis implicates one or more of the information sources extracted by the algorithm from the preprocessor input stream. Once identified, the information source, and that portion of the algorithm which uses it, must be evaluated to determine how reliable the information source is and how the information it contains can be used more effectively. We therefore add:

g. For each information source used in LISTEN, has that source been adequately modeled and does LISTEN make the most effective use of the information it carries, consistent with real-time operation?

Finally, the possible existence of information sources in the preprocessor output stream which have been overlooked, and which may contribute significantly to disambiguation, must be recognized. Therefore we also add:

h. Are there sources of information in the preprocessor output stream currently unused by LISTEN, and which should be incorporated into the recognition algorithm?

NAVTRAEQUIPCEN 77-C-0096-1

DISTRIBUTION LIST

Naval Training Equipment Center 43
Orlando, Florida 32813

Defense Documentation Center 12
Cameron Station
Alexandria, VA 22310

Chief of Naval Operations (OP-39) 1
Department of the Navy
(Mr. W. H. Primas)
Washington, DC 20360

Seville Research Corp. 1
Suite 400 Plaza Bldg
Pace Blvd at Fairfield
Pensacola, FL 32505

USAHEL/AVSCOM 1
Dir, RD&E
ATTN: DRXHE-AV (Dr. Hofmann)
P. O. Box 209
St Louis, MO 63166

Army Training Support Center 1
Attn: LTC Pike
Ft Eustis, VA 23604

Commandant 1
USA Field Artillery School
Target Acquisition Dept
ATTN: Eugene C. Rogers
Ft Sill, OK 73503

Chief of Naval Research 1
Code 458
Dept of Navy
Arlington, VA 22217

Chief of Naval Research 1
Psychological Sciences
Code 450, Dept of Navy
Arlington, VA 22217

Chief of Naval Operations 1
OP-987H, Dept of Navy
ATTN: Dr. R. G. Smith
Washington, DC 20350

Library 1
Navy Personnel Research and
Development Center
San Diego, CA 92152

Director 1
Defense Research and Engineering
ATTN: LCOL H. Taylor, OAD E&LS
Washington, DC 20301

Commander 1
Naval Sea Systems Command (047C1)
Washington, DC 20362

D.W.T. Naval Ship R&D Ctr 1
Code 1822, T. Rhodes
Bethesda, MD 20084

Commander 1
Naval Sea Systems Command
Code 03416 (Mr P. J. Andrews)
Washington, DC 20360

US Air Force Human Resources 1
Lab/DOJZ
Brooks AFB, TX 78235

ASD SD24E 1
ATTN: Mr. Harold Kottmann
Wright-Patterson AFB, OH 45433

Commander 1
Navy Air Force, US Pacific Fleet
NAS North Island (Code 316)
San Diego, CA 92135

Chief 1
ARI Field Unit
P. O. Box 2086
Fort Benning, GA 31905

Chief 1
Naval Education & Training Liaison Off
AF Human Resources Laboratory
Flying Training Div
Williams AFB, AZ 85224

Commander 1
Naval Air Systems Command
Naval Air Systems Command Headquarters
(AIR 413-B)
Washington, DC 20361

Naval Weapons Center 1
Code 3143
ATTN: Mr. George Healey
China Lake, CA 93555

Dr. Jesse Orlansky
Institute for Defense Analyses
Science & Technology Div
400 Army-Navy Drive
Arlington, VA 22202 1

Director Educational Development
Academic Computing Center
U. S. Naval Academy
Annapolis, MD 71402 1

Mr. John P. Burg
President, Time & Space
Processing, Inc.
10430 N. Tantau Ave
Cupertino, CA 95014 1

Mr. Franklin S. Cooper
Associate Director of Research
Haskins Laboratories, Inc.
270 Crown St.
New Haven, CT 06511 1

CDR P. M. Curran
Naval Air Development Center,
Code 6041
Aircraft & Crew Systems Tech. Dir.
Warminster, PA 18974 1

Mr. William Dewing
Manufacturing Research Engineer, Sr.
Lockheed Missiles & Space Co.
Box 504, 0/86-76, B/182
Sunnyvale, CA 94086 1

Mr. N. Rex Dixon
Speech Processing Consultant
IBM, Thomas J. Watson Research Ctr
PO Box 218
Yorktown Heights, NY 10598 1

Dr. G. R. Doddington
Speech Systems Research
Systems & Information Sciences Lab
Texas Instruments, Inc.
PO Box 5012, M/S 5
Dallas, TX 75222 1

Mr. Emmett L. Herron
Human Factors Engineer
Bunker Ramo Corporation
4130 Linden Ave., Suite 302
Dayton, OH 45432 1

Mr. Horace Enea
President, Heuristics, Inc.
900 N. San Antonio Rd
Suite C-1
Los Altos, CA 94022 1

Mr. Leon A. Ferber
Vice President, Perception Tech. Corp
95 Cross St
Winchester, MA 08190 1

Hallie M. Funkhouser
Technical Assistant
NASA, Ames Research Center
Mail Stop 239-3
Moffett Field, CA 94035 1

Mr. Charles W. Geer
Engineer, The Boeing Co.
PO Box 2999, M.S. 13-85 ORG 2-3541
Seattle, WA 98124 1

CDR D. C. Hanson
Director, Electromagnetic Technology
ONR Code 221
800 N. Quincy St
Arlington, VA 22217 1

LT Steve Harris
Naval Aerospace Medical Research Lab
Pensacola, FL 32508 1

Mr. Marvin B. Herscher
Executive Vice President
Threshold Technology, Inc.
1829 Underwood Blvd.
Delran, NJ 08075 1

Lt Col Robert L. Hilgendorf, USAF
Aeronautical Systems Div/AERS
Wright-Patterson AFB
Dayton, OH 45433 1

LCDR Norman E. Lane
Naval Air Development Ctr, Code 6041
Warminster, PA 18974 1

Dr. Wayne A. Lea
Research Linguist
Speech Communications Research Lab
800A Miramonte Dr
Santa Barbara, CA 93109 1

Dr. Barry M. Leiner 1
Senior Development Engr,
Probe Systems, Inc
655 N. Pastoria Ave
Sunnyvale, CA 94086

Mr Leon Lerman 1
Lockheed Missiles & Space
PO Box 504, Dept 86-10, Bldg 153
Sunnyvale, CA 94080

Mrs. Beatrice Oshika 1
System Development Corp
2500 Colorado Ave
Santa Monica, CA 90406

Mr. Warren Lewis 1
Human Engineering Branch
Naval Ocean Systems Ctr, Code 8231
San Diego, CA 92152

Mr. Arthur W. Lindberg 1
Electronics Engr, US Army Avionics
R&D Activity, DAVAA-S
Ft Monmouth, NJ 07703

Mr. Bruce T. Lowerre 1
Computer Scientist, Systems Control,
Inc
1801 Page Mill Rd
Palo Alto, CA 94304

Capt Barry P. McFarland 1
USAF, ASD/ENECH
Wright-Patterson AFB
Dayton, OH 45433

Mr. Don F. McKechnie 1
Research Psychologist
Aerospace Medical Research Lab
Human Engineering Div
Wright-Patterson AFB
Dayton, OH 45433

Mr. John D. Markel 1
President, Signal Technology, Inc
15 W. De La Guerra
Santa Barbara, CA 93101

Capt Ronald J. Marini, USAF 1
ASD/AER-EX
Wright-Patterson AFB
Dayton, OH 45433

Mr. Thomas B. Martin 1
President, Threshold Technology, Inc.
1829 Underwood Blvd
Delran, NJ 08075

Mr. John Martins, Jr. 1
Project Engr, Naval Underwater Sys Ctr
New London Laboratory MC 315
New London, CT 06320

Dr. Mark F. Medress 1
Manager, Speech Communications
Sperry Univac Defense Systems
Speech Communications Dept
Univac Park, PO Box 3525 - UOP16
St Paul, MN 55165

LT Thomas M. Mitchell 1
Naval Air Development Center, Code 604
Warminster, PA 18974

Mr. Don Murray 1
Telcom Systems, Inc
320 West Street Rd
Warminster, PA 18974

Mr. J. Michael Nye 2
President, Marketing Consultants
International, Inc.
100 W. Washington St, Suite 216
Hagerstown, MD 21740

Mr. Richard W. Obermayer 1
Navy Personnel R&D Ctr, Code 34
San Diego, CA 92152

Mr. Bob O'Hagan 1
Staff Scientist, Bell Northern
Research
3174 Porter Dr.
Palo Alto, CA 94304

Mr. M. Ohkohchi 1
IBM, Japan LTD
Tokyo Scientific Center
No. 21 Mori Building
1-4-34 Roppongi,
Minato-Ku Tokyo, 106

Mr. Robert Osborn 1
VP Engineering, Dialog Systems, Inc.
32 Locust St
Belmont, MA 02178

LT Jerry M. Owens 1
Chief, Engineering Psychology Div
Naval Aerospace Medical Research Lab
Aerospace Psychology Dept
Pensacola, FL 32508

Mr. Thomas W. Page 1
Director, National Security Agency
9800 Savage Rd
Attn: R-54 Page
Ft George G. Meade, MD 20755

Mr. Ernest E. Poor 1
Naval Air Systems Command
Code NAIR 413B, Room 336
Washington, DC 20361

Dr. Raj Reddy 1
Professor, Dept of Computer Science
Carnegie-Mellon Univ
Pittsburgh, PA 15213

Dr. June E. Shoup 1
Dir, Speech Communications Research Lab
Lab
800A Miramonte Dr
Santa Barbara, CA 93109

Mr. Lon Sorenson 1
Systems Engr, SEMCOR, Inc.
Strawbridge Lake Office Bldg, Rt 38
Moorestown, NJ 08057

Mr Sverre Nils Straatveit 1
Electronics Engr, Naval Underwater Systems Center, Code 315
New London, CT 06320

Mr. Melvin L. Strieb 1
Program Manager, Human Factors Analytics
2500 Maryland Rd
Willow Grove, PA 19090

Mr. Sam S. Viglione 1
Interstate Electronics
707 E. Vermont Ave
Anaheim, CA 92803

Dr. Donald E. Walker 1
Senior Research Linguist
SRI International
Menlo Park, CA 94025

Mr. I. James Whitton 1
Systems Engr, General Electric - AES
831 Broad St. MD700
Utica, NY 13503

Mr. Jared J. Wolf 1
Sr Scientist, Bolt, Beranek & Newman, Inc
50 Moulton St
Cambridge, MA 02138

Mr. Kenneth R. Woodruff 1
Sr Scientist - Human Factors
Systems Research Laboratories, Inc
2800 Indian Ripple Rd
Dayton, OH 45440

Mr. Robert S. Hartman 1
VP Electronics, Gould, Inc.
Hydrosystems Division
125 Pinelawn Road
Melville, NY 11746

Aerospace Psychology Dept 1
Naval Aero Med Research Lab (L522)39
NAS
Pensacola, FL 32512

Commander 1
Training Command
Attn: Education Advisor
US Pacific Fleet
San Diego, CA 92147