AD-A013 808

STOCHASTIC MODELING AS A MEANS OF AUTOMATIC SPEECH RECOGNITION

CARNEGIE-MELLON UNIVERSITY

PREPARED FOR
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

APRIL 1975

STOCHASTIC MODELING AS A MEANS OF

AUTOMATIC SPEECH RECOGNITION

James K. Baker

April 1975

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR - TR - 75 - 1094	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) STOCHASTIC MODELING AS A MEANS OF AUTOMATIC SPEECH RECOGNITION		5. TYPE OF REPORT & PERIOD COVERED Interim 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James K. Baker		8. CONTRACT OR GRANT NUMBER(s) F44620-73-C-0074
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Computer Science Dpt Pittsburgh, PA 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61101D A02466
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd Arlington, VA 22209		12. REPORT DATE April, 1975 13. NUMBER OF PAGES 114
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Air Force Office of Scientific Research/NM 1400 Wilson Blvd Arlington, VA 22209		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)

PRICES SUBJECT TO CHANGE

18. SUPPLEMENTARY NOTES

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

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

Automatic recognition of continuous speech involves estimation of a sequence X(1), X(2), X(3), ... , X(T) which is not directly observed (such as the words of a spoken utterance), based on a sequence Y(1), Y(2), Y(3), ... , Y(T) of related observations (such as the sequence of acoustic parameter values) and a variety of sources of knowledge. Formally, we wish to find the sequence x[1:T] which maximizes the *a posteriori* probability Pr(X[1:T]=x[1:T] / Y[1:T]=y[1:T], A, L, P, S), where A, L, P, S represent the acoustic-phonetic, lexical, phonological, and syntactic-semantic knowledge. A speech recognition system must attempt to approximate a solution to this problem, whether or not the system uses a formal stochastic model.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1

Block 20/Abstract

The DRAGON speech recognition system models the knowledge sources as probabilistic functions of Markov processes. The assumption of the Markov property allows the use of an optimal search strategy. The DRAGON system finds the sequence $x[1:T]$ which maximizes the above probability, as given by the Markov model. In effect, the system searches all possible sentences in the grammar, all possible pronunciations of each sentence, and all possible dynamic time warpings of each such phonetic string to best fit it to the acoustic observations. This optimal search is carried out by the procedure expressed in equations (1) and (2).

$$(1)\quad \gamma(t,j) = \mathrm{Max}_i \{\, \gamma(t-1,i)\Pr(\, X(t)=j \mid X(t-1)=i,\, A,L,P,S\,) \\ \Pr(\, Y(t)=y(t) \mid X(t-1)=i,\, X(t)=j,\, A,L,P,S\,)\,\}$$

Let $I(t,j)$ be any value of i for which the above maximum is achieved.

$$(2)\quad x(t) = I(t+1,\, x(t+1))$$

The use of a general theoretical framework, with an explicit representation for the solution process, greatly simplifies the speech recognition system. Equations (1) and (2) represent the entire recognition process. Despite its simplicity the system can, to some degree, use knowledge from each of the domains A,L,P, and S.

A simplified implementation of the DRAGON system has been developed using knowledge A and L, and some of the knowledge from S. This implementation has been tested on 102 utterances from 5 interactive computer tasks. The size of the integrated Markov network representing the knowledge sources is 410, 702, 916, 498, and 2356 states, respectively, for the 5 tasks whose vocabulary sizes are 24, 66, 37, 28, and 194 words, respectively, and which have grammars of varying degrees of complexity. The time required for recognition of an utterance is proportional to the length of the utterance and is given approximately by the expression (recognition time) = (utt length)(20.9 + .067(net size)). Since a complete optimal search is performed, the recognition time is independent of the amount of noise in the signal or the number of errors in intermediate recognition decisions. The system correctly recognized 49% of the utterances and correctly identified 83% of the 578 words

i-b

STOCHASTIC MODELING AS A MEANS OF AUTOMATIC SPEECH RECOGNITION

James K. Baker

April 1975

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Speech and Computer Science.

Mellon Institute of Science
Carnegie-Mellon University
Pittsburgh, PA 15213

This work was supported by the Defense Advanced Research Projects Agency under contract F44620-73-C-0074 and is monitored by the Air Forec Office of Scientific Research.

i-C

STOCHASTIC MODELING AS A MEANS OF AUTOMATIC SPEECH RECOGNITION

James K. Baker
Carnegie-Mellon University

Automatic recognition of continuous speech involves estimation of a sequence **X**(1), **X**(2), **X**(3), ... , **X**(T) which is not directly observed (such as the words of a spoken utterance), based on a sequence **Y**(1), **Y**(2), **Y**(3), ... , **Y**(T) of related observations (such as the sequence of acoustic parameter values) and a variety of sources of knowledge. Formally, we wish to find the sequence x[1:T] which maximizes the *a posteriori* probability $\Pr(\mathbf{X}[1{:}T]=x[1{:}T] \mid \mathbf{Y}[1{:}T]=y[1{:}T], \mathbf{A}, \mathbf{L}, \mathbf{P}, \mathbf{S})$, where **A**, **L**, **P**, **S** represent the acoustic-phonetic, lexical, phonological, and syntactic-semantic knowledge. A speech recognition system must attempt to approximate a solution to this problem, whether or not the system uses a formal stochastic model.

The DRAGON speech recognition system models the knowledge sources as probabilistic functions of Markov processes. The assumption of the Markov property allows the use of an optimal search strategy. The DRAGON system finds the sequence x[1:T] which maximizes the above probability, as given by the Markov model. In effect, the system searches all possible sentences in the grammar, all possible pronunciations of each sentence, and all possible dynamic time warpings of each such phonetic string to best fit it to the acoustic observations. This optimal search is carried out by the procedure expressed in equations (1) and (2).

$$(1)\quad \gamma(t,j) = \mathrm{Max}_i \{ \gamma(t-1,i)\Pr(\mathbf{X}(t)=j \mid \mathbf{X}(t-1)=i, \mathbf{A},\mathbf{L},\mathbf{P},\mathbf{S}) \\ \Pr(\mathbf{Y}(t)=y(t) \mid \mathbf{X}(t-1)=i, \mathbf{X}(t)=j, \mathbf{A},\mathbf{L},\mathbf{P},\mathbf{S}) \}$$

Let I(t,j) be any value of i for which the above maximum is achieved.

$$(2)\quad x(t) = I(t+1, x(t+1))$$

The use of a general theoretical framework, with an explicit representation for the solution process, greatly simplifies the speech recognition system. Equations (1) and (2) represent the entire recognition process. Despite its simplicity the system can, to some degree, use knowledge from each of the domains **A**,**L**,**P**, and **S**.

A simplified implementation of the DRAGON system has been developed using knowledge **A** and **L**, and some of the knowledge from **S**. This implementation has been tested on 102 utterances from 5 interactive computer tasks. The size of the integrated Markov network representing the knowledge sources is 410, 702, 916, 498, and 2356 states, respectively, for the 5 tasks whose vocabulary sizes are 24, 66, 37, 28, and 194 words, respectively, and which have grammars of varying degrees of complexity. The time required for recognition of an utterance is proportional to the length of the utterance and is given approximately by the expression (recognition time) = (utt length)(20.9 + .067(net size)). Since a complete optimal search is performed, the recognition time is independent of the amount of noise in the signal or the number of errors in intermediate recognition decisions. The system correctly recognized 49% of the utterances and correctly identified 83% of the 578 words.

ACKNOWLEDGEMENTS

I wish to thank Leonard Baum, who introduced me to the theory of a probabilistic function of a Markov process, Raj Reddy, who guided my research in speech recognition, and Janet MacIver Baker, who introduced me to the problem of speech recognition and who made it all worthwhile. This research was supported in part by the Advanced Research Projects Agency of the Department of Defense under contract no. F44620-73-C-0074 and monitored by the Air Force Office of Scientific Research. The final editing of the dissertation was done while the author was with the Speech Processing Group, Computer Science Department, IBM Thomas J. Watson Research Center.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Speech recognition, a task which humans do efficiently and well, is very difficult to do by automatic procedures. There is a great deal of ambiguity in the actual acoustic signal—ambiguity which can be resolved only by applying other sources of knowledge in addition to the acoustic signal([A1], [R7], [N2]). In recent years much research has been devoted to developing the other sources of knowledge that are available in analyzing speech which is restricted to a specialized domain of discourse([R4], [R5], [T1], [D1], [P2], [W3], [F2], [B6], [W1], [L1], [J3]). In such a specialized domain there is generally a restricted vocabulary, so one source of knowledge is the **lexical** knowledge. The utterances are constrained to be grammatical and sometimes the grammar is a special restricted one, so there is **syntactic** knowledge. In some of the systems the specialized domain is an interactive task with the computer as a participant. Thus there is an operational definition of whether an utterance is "meaningful" (that is, can the computer interpret the utterance in relation to the interactive task), and therefore there is a kind of **semantic** knowledge([R6]).

In order to apply these sources of knowledge in speech recognition, it is necessary to represent this knowledge in a form that can be compared with the acoustic observations. There are two operations which are essential in any speech recognition system: searching and matching. Suppose one knowledge source, such as syntax, hypothesizes a word or a sequence of words. This hypothesis can only be verified by matching the words with the events observed by the other sources of knowledge, such as the actual acoustic signal. A **matching** procedure is needed to evaluate any particular hypothesis. A **searching** procedure is needed to explore the space of possible hypotheses.

SEARCHING AND MATCHING IN SPEECH RECOGNITION SYSTEMS

The various speech recognition systems which have been developed use a great variety of searching and matching procedures and employ them in many different ways. The **DRAGON** speech recognition system, the subject of this thesis, is based on a systematic use of a particular abstract model to represent many of the sources of knowledge needed for speech recognition. This uniformity of representation then allows a powerful general searching/matching technique to be applied to the speech recognition system as a whole. First let's consider some of the ways in which searching and matching procedures are used in other speech recognition systems.

The HEARSAY I system ([E2], [R3], [R4], [R5]) employs a hypothesize and test paradigm. There is a separate programming module for each source of knowledge which is represented. Each module is responsible for generating hypotheses based on its own internal knowledge. Each hypothesis is then verified by each of the modules (that is, each module matches the hypothesis against its own knowledge) and a combined rating is computed. The modules communicate with each other primarily by stating hypotheses about the sequence of words and each module has its own matching procedures for relating such "word-level" hypotheses to its own specialized knowledge. The search strategy is basically a best-first tree search. Words are hypothesized proceeding left-to-right in the utterance. At any point in the analysis new hypotheses are generated which are extensions of the best partial sequence of words obtain so far in the analysis. On the next round of the analysis, either the best such extension becomes the best partial sequence or, if all such extensions get sufficiently low ratings, a previous partial sequence (which had been the second best partial sequence) is reactivated.

In the HEARSAY II system ([L2]) the matching and search mechanisms are much more general and flexible. Hypotheses are not restricted to the word level, but instead are organized into an indefinite number of levels ranging from sub-phonetic acoustic segements to semantics and pragmatics. There are a large number of independent knowledge source modules. Each knowledge source repeatedly applies matching procedures to compare the data structure of existing hypotheses with its internal knowledge base. Whenever a match is found the knowledge source takes the appropriate action to add an hypothesis or otherwise modify the data structure. The search strategy consists of scheduling which knowledge sources get activated and in what order, based on a variety of scores and ratings for the hypotheses that are in the data structure at a given time.

In the Automatic Recognition of Continuous Speech (ARCS) systems ([D1], [T1], [T2], [T3], [P1], [P2], [R1]) a variety of tests are applied to the acoustic signal to derive a (noisy) phonetic string and there is a language model for generating sequences of words. The conversion of the noisy phonetic string to an orthographic string is then performed by searching and matching procedures. For each word there is a network representing all permitted pronunciations of the word. The conditional probability of a particular word producing a given phonetic string can be computed explicitly, and is used to measure the degree of match. The search procedure is a best-first tree search implemented by a sequential decoding algorithm. Earlier versions of the ARCS system had the same general structure, but performed the matching at the phonetic level rather than at the word level.

The knowledge sources in the SPEECHLIS system ([B7], [N1], [R9], [W2], [W3]) represent their information in lattice structures which show all the alternatives at any point in time. The word-lattice is generated by matching each lexical item with the entries in the segment lattice. A semantic component searches the word lattice to develop "theories" of semantically related words. The semantic component continues to work on the theories with the greatest likelihood scores. When the semantics component can add no more words to a theory, the theory is passed to a syntax component which performs a parse and fills in any gaps.

The CASPER system ([F2], [K1]) performs a match between lexical items and a noisy phonetic sequence by using multiple dictionary entries, phonological rules embedded in the dictionary, and a "degarbling" procedure. The search is controlled by an augmented context-free grammar which performs a left-to-right, bottom-up parse.

The Vocal Data Management System ([B6], [R8]) developed at SDC employs a strategy of "Predictive Linguistic Constraints." The parser attempts to predict phrases based on a simple user model, thematic patterning, and grammatical and semantic constraints. Fixed directional parsing is replaced by a more general approach so that processing may be initiated at any point in the utterance. Lexical items are matched against the acoustic-phonetic data by a word mapper and a syllable mapper. The word mapper handles alternate pronunciations of a word, decides likely times for syllable boundaries, and checks for co-articulation effects across syllable boundaries. The syllable mapper compares a syllable candidate with the sequence of acoustic parameters.

The SRI Speech Understanding System ([P3], [P4], [W1]) uses a special "word function" for each item in the lexicon. Each word function consists of a series of Fortran subroutines that look for a match between its particular word and data from a variety of sources based on parameters extracted from the acoustic signal. The parser executes a top-down, "best-first" strategy. In addition to its parsing function, it calls on the other components and coordinates information among them.

The Univac Speech Understanding System ([L1]) uses a prosodically-guided strategy. Prosodic features are used to break sentences into phrases, locate the stressed syllables within those phrases, and guide procedures for both phone classification and higher level linguistic analysis. This strategy requires a search procedure which is able to initiate processing at any point in the utterance as indicated by the prosodic features. Specific search and matching procedures have not yet been implemented for this system.

The speech recognition system being developed at the IBM Watson Research Center ([B1], [J3]) is based on a linguistic sequential decoder. The decoder consists of four major subparts: 1) a statistical model of the language, 2) a phonemic dictionary and statistical phonological rules, 3) a phonetic matching algorithm, 4) word level search control. The search procedure is a stack decoding algorithm which seeks that word sequence which has the maximum *a posteriori* probability, conditional on the language and the observed acoustic sequence. Statistical matching is done between hypothesized words and a noisy phonetic string obtained by acoustical analyses.

Even these greatly simplified descriptions make it clear that there is a great variety of ways in which searching/matching strategies can be implemented. However, certain common features can be distinguished. Most of the systems perform matching only at one level. Generally the matching is between lexical items and a noisy phonetic string (ARCS, SPEECHLIS, CASPER, IBM-Watson). Thus for example, in these systems, words and phrases are not directly matched to the acoustics. For most of the systems, the search is controlled primarily at the word level (HEARSAY I, ARCS, SPEECHLIS, CASPER, SDC, SRI, IBM-Watson). Only two systems (ARCS, IBM-Watson) have explicit statistical models from which to derive matching scores.

In addition to the general purpose searching/matching which is usually used in transforming a noisy phonetic string to a word string, several specialized procedures are used. SDC has a mapping between syllables and acoustic parameters. SRI matches words directly with acoustics. The early ARCS system matched the language directly onto the noisy phonetic string. The segment data in the SPEECHLIS system is a lattice of alternatives, so matching even a single lexical item involves a small lattice search. Each of the modules in the HEARSAY systems includes specialized matching procedures.

FEATURES OF THE DRAGON SYSTEM

The fundamental idea behind the **DRAGON** system is that each of the knowledge sources can be represented by a single, general, abstract model. Then powerful general search/match algorithms can be employed without worrying about all the special characteristics of each individual knowledge source. These special characteristics are not ignored, but they get incorporated into the data structures and not into the searching/matching procedures. The model which is used throughout the **DRAGON** system is that of a probabilistic function of a Markov process[B8].

The sequence of random variables $Y(1), Y(2), Y(3), \ldots, Y(T)$ is said to be a probabilistic function of a Markov process if there is a sequence of random variables $X(1), X(2), X(3), \ldots, X(T)$ such that the sequences of **X**'s and **Y**'s satisfy equations (5) and (6) of Chapter II. The techniques for analyzing such a system are described in Chapter II. The interpretation is that the **Y**'s are a sequence of random variables that we observe and which depend probabilistically on the **X**'s which we do not observe. We wish to make inferences about the values of the **X**'s from the observed values of the **Y**'s. Chapter III describes how the knowledge sources in a speech recognition system can be represented in terms of this type of model. Chapter IV describes a simplified implementation of these ideas. Performance results are given which show that even this greatly simplified implementation is a complete and powerful speech recognition system.

The important features of the **DRAGON** system are:

1) Generative form of model;

2) Hierarchical arrangement of knowledge sources;

3) Integrated network representation;

4) General theoretical framework;

5) Optimal stochastic search.

In comparing the features of different speech recognition systems, attention is often focused on the control structures and the methods of communication among the knowledge source modules. Thus a system might be characterized by whether the analysis proceeds top-down or bottom-up (or some mixture), whether there is a best-first tree search or some other control mechanism, and whether the analysis proceeds in a strict left-to-right fashion or can start at any point in the utterance. For several reasons, the **DRAGON** system cannot be easily characterized by these conventional dichotomies, so the discussion of them is postponed until the major features of the system are described.

(1) Generative form of the model

The generative form is a natural one for a probabilistic function of a Markov process. Generative rules are formulated as conditional probabilities. For example, if we know which phone occurs at a given time, vocal tract models allow us to predict the values of the acoustic parameters. That is, a conditional probability distribution is defined in acoustic parameter space. If we know which word occurs during a given segment of time, phonological rules allow us to estimate the probability of various phone sequences representing different pronunciations of the word. A statistical model for the errors of an automatic phone classifier allows us to calculate the probability of the classifier producing a specific sequence of labels, conditional on the true sequence of phones being a particular phone sequence. The grammar for a specific task domain produces a conditional probability distribution in the space of word sequences such that ungrammatical sequences have zero probability.

Each of the knowledge sources in the **DRAGON** system is represented in a generative form as a probabilistic function of a Markov process. However, Bayes' theorem allows the computation to be performed analytically. The model tells the conditional probability of producing a specific sequence of acoustic parameter values from a specific sequence of words. Applying Bayes' theorem, we can compute the *a posteriori* probability of a sequence of words from the observed sequence of acoustic parameter values.

(2) Hierarchical arrangement of knowledge sources

The sources of knowledge are organized into a hierarchy based on the following observation: The "higher" levels of a speech recognition system change state less frequently than the "lower" levels. Thus a single syntactic-semantic state corresponds to a sequence of several words; a single word corresponds to a sequence of several phones; and a phone corresponds to a sequence of acoustic parameter values. The hierarchy is not absolute—for example, syntax and semantics are together a single multi-level process—but it provides a convenient means for combining the Markov processes which represent the individual sources of knowledge.

To see how the knowledge can be represented as a hierarchy of generative models, let's consider a simplified example. Consider a language with only two sentences: "What did you see?" and "Where did you go?" At the word level this language can be represented by the network shown in Figure 1.

GRAMMAR NETWORK

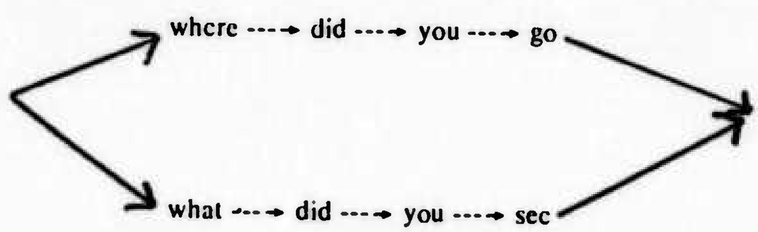


FIGURE 1

This model is generative in the sense that if we know a partial sequence of words (e.g. "What did") the model tells exactly which word can come next ("you"). But we do not directly observe the words (we only observe the associated acoustic events), so we must compute the *a posteriori* probability of any word sequence using the techniques of Chapter II.

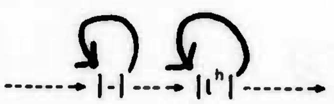
WORD NETWORK

-------→ /w/ ---→ /ʌ/ ---→ /t/ -------→

FIGURE 2

In the next lower level of the hierarchy we represent the relationship between the words and the phones. To keep the network simple, only a single pronunciation is represented for each word. For example, the network for "what" is shown in Figure 2. It is also possible to add another level to the hierarchy connecting the phones to the expected acoustic parameter values. The stop consonants and the dipthongs are broken up into several sub-phonemic segments. The network for $[t^h]$ is shown in Figure 3. The connection with acoustic parameters is then represented by a table giving the statistical distribution of parameter values for each type of segment. Phonological and acoustic-phonetic rules, which are omitted from this example, could be represented either at the broad phonetic level (such as, if the /t/ is flapped) or at the acoustic segment level (whether the /t/ is released and its degree of aspriation, if released).

PHONE NETWORK



(where - represents the pause portion, and t^h represents the release/aspiration)

FIGURE 3

The nodes in Figure 3 have arcs which point back to themselves because we are representing two processes which are asynchronous with respect to each other. That is, the acoustic parameters are measured at fixed time intervals (say once every 10 milliseconds), but each sub-phonemic acoustic segment lasts for an unknown period of time. So, if we time our stochastic process at one step every 10 milliseconds, then the process may stay in the same state for several units of time, as indicated by an arc returning to the same node. A phone which consists of a single acoustic segment is represented be a phone network with a single node, but with a loop from the node back to itself, again indicating that the process may stay in this state for several units of time.

(3) Integrated network representation

To describe a point in the hierarchical state space, we must describe its position in a network at each level of the hierarchy. For example, the description (1) "the pause segment" of (2) "the [t^h]" of (3) "the word 'what'," descibes a particular point in the hierarchical state space in our simple example. Since each of the networks is finite, it is possible to define a new network with a separate node for each point in the hierarchical space. In terms of the knowledge represented, this new network and the hierarchy of networks are equivalent. The change is primarily one of convenience. The integrated network representing our simplified example is shown in Figure 4.

INTEGRATED NETWORK

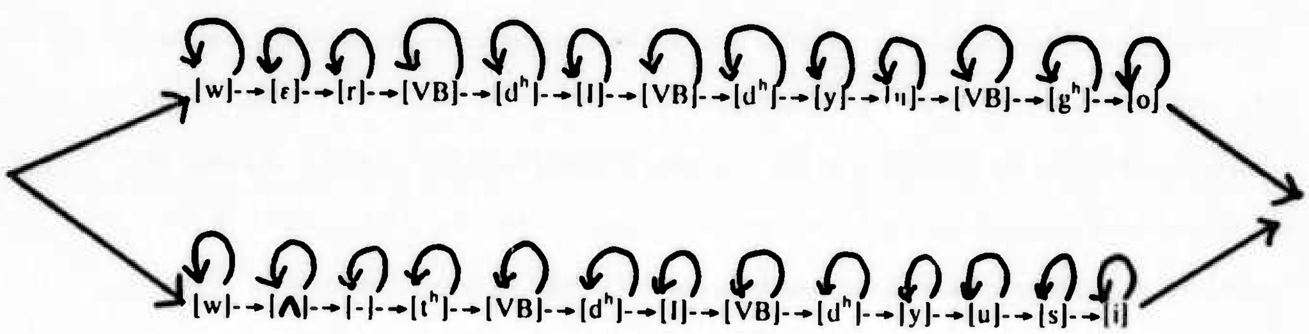


FIGURE 4

Actually it is possible to represent more knowledge in the integrated network than in the hierarchical system. For example, phonological rules which apply across word boundaries (such as the palatalization in the word pair "did you") may be used to make modifications to the network. Note that the integrated network, because it is derived in a special way from a hierarchy, is very sparse. In the example each node (except the end nodes) is connected to (has an arc pointed toward) only itself and one other node. Even with a more general language and networks representing phonological rules, almost any node that is not adjacent to a word boundary would be connected only to itself and one, two, or three other nodes. Thus, in a network with thousands of nodes, there are only two or three arcs per node (instead of the thousands which would be possible). This property of sparseness has implications for the implementation of the speech recognition system, as is discussed in Chapters II and IV.

The size of the integrated network for a given task depends on the vocabulary size, the complexity of the grammar, and on some of the details of the implementation. The five tasks discussed in Chapter IV have vocabulary sizes of 24, 66, 37, 28, and 194 words, respectively. The number of nodes in the integrated network is 410, 702, 916, 498, and 2356, respectively. Even the largest network is small enough so that the recognition system described in Chapter IV can keep all of its intermediate computational results in the computer's core memory with no need to use secondary storage.

Note that we go from a group of separate knowledge sources to an integrated network representation in essentially three steps. First, each knowledge source is represented as a probabilistic function of a Markov process. The details of this step are described in Chapter III. In this chapter the skeleton of the idea is exposed by way of the associated network. Second, the knowledge sources are arranged in a hierarchy. In a sense, it is this step which is crucial. It relies on the special relationships among the knowledge sources for speech recognition systems. It would not necessarily be applicable to knowledge sources for other problems even if the knowledge sources are representable as probabilistic functions of a Markov process. Third, the hierarchy of networks is converted into an equivalent single network (and the hierarchy of Markov processes is replaced by a single Markov process). Athough this final step changes the apparent external structure of the system, it does not change the substance.

(4) General theoretical framework

As stated before, the **DRAGON** system relies throughout on a particular abstract model—that of a probabilistic function of a Markov process. A sequence of random variables Y(1), Y(2),

$Y(3), \ldots, Y(T)$ is said to be a probabilistic function of the Markov process $X(1), X(2), X(3), \ldots, X(T)$ if these random sequences satisfy equations (5) and (6) of Chapter II. These equations may be paraphrased as requiring that, for any t, $X(t)$ depends only on $X(t-1)$ and $Y(t)$ depends only on $X(t)$ and $X(t-1)$. Chapter III describes how various knowledge sources may be represented by such a model.

The formulas that the model produces are similar to the formulas used in other statistically based speech recognition systems (ARCS and IBM-Watson). In certain ways, either system can be considered as a special case of the other. The difference is more one of emphasis than one of kind. The emphasis in the **DRAGON** system is one of representing each of the knowledge sources in a uniform theoretical framework. Thus specialized procedures for handling the data for a particular knowledge source are avoided.

The only specialized procedures are those used in setting up the integrated network to represent the combined knowledge sources. In recognizing a particular utterance, the only procedure which is used is one which is based only on the general properties of a probabilistic function of a Markov process. For example, the type of specialized procedure which is absent is one which would take acoustic parameters and with a complicated set of rules, thresholds, and decisions produce a raw phonetic string intended to be as close as possible to a phonetic transcription of the utterance. As explained in Chapter III, if such a procedure is available, the **DRAGON** system can use the phonetic string which is produced. But on the other hand, if such a procedure is not used, the **DRAGON** system can operate directly on the acoustic parameters, since the acoustic-phonetic knowledge can be represented as a probabilistic function of a Markov process and be incorporated into the hierarchy.

(5) Optimal stochastic search

The Markov model used in the **DRAGON** system requires a finite state space. In that sense it is less general than the augmented network systems (SPEECHLIS, CASPER, SRI) and stack decoding statistical systems (ARCS, IBM-Watson). However, a large finite network can represent most of the important information and some of the things which it cannot represent are irrelevant in a recognition problem in which the input is a noisy phonetic string with arbitrary insertions and deletions. The finite state space and the Markov model make possible the powerful algorithms which are described in Chapter II.

The search algorithm of the **DRAGON** system is unique in that rather than search a tree (the tree of possible word sequences) one branch at a time in some best-first or depth-first manner, it searches the entire space of all possible paths through its network. All paths of a given length are, in effect, searched in parallel. At the end of the analysis a path is obtained which is an optimum over all possible paths through the network. This path represents that interpretation of an utterance which, among all possible interpretations, best matches the given observed values of the acoustic parameters.

To search this entire space may seem to be drastic, but with the Markov model and the algorithms of Chapter II, it can be done very efficiently. These algorithms are not new. The inductive computation of the best partial sequence, as done by equation (18) of Chapter II, is an application of dynamic programming to the general network search problem([**B9**]). It corresponds to an algorithm used in communications and coding theory, known as the Viterbi algorithm([**V1**]). There are other algorithms for sequential decoding([**F1**], [**J1**], [**J2**]), which are also based on maximizing the *a posteriori* probability according to such a stochastic model, and several of them have been successfully applied to speech recognition (ARCS and IBM-Watson).

The number of computations required to search the space of all possible paths through the network is proportional to (the length of the utterance) times (the number of arcs in the network). For a given network, the computation time is linear in the length of the utterance and is independent of the amount of noise or the number of errors in any input string. This property is in sharp contrast to depth-first or best-first algorithms for which there is no effective upper bound for the amount of computation (except a search of the entire tree, one branch at a time). The sequential search algorithms do, in fact, occasionally need to be terminated before completion of the analysis because they exhaust the available time or storage.

On the other hand, although the Markov model permits a complete optimum search in a time that is linear in the length of the utterance, the proportionality factor is large, especially for large vocabularies. Many things could be done to reduce the computation time required by the **DRAGON** system, and they are an important and interesting area for future research, but in the work reported in this thesis there has been no attempt to minimize the computation time. Lowerre ([L3]) has rewritten the **DRAGON** program to execute much faster with no change in recognition results. The computation times given in Chapter IV, therefore, should be regarded as an upper bound on the amount of time required by the techniques presented in this thesis and as a demonstration that complete optimal search is not impossible.

The **DRAGON** system cannot be characterized as either top-down or bottom-up because it has aspects of both types of system. The models are given in a generative form, which is normal for top-down systems. However, by applying Bayes' formula the analysis proceeds in the analytic rather than the synthetic direction. But even more significant is the fact that the integrated representation makes it impossible to distinguish whether the acoustic knowledge is helping to direct the syntactic analysis, or if the syntactic knowledge is helping to direct the acoustic analysis. Instead of a system with separate components with specific feed-back and feed-forward mechanisms for transmitting information, the system is completely integrated.

The **DRAGON** system represents an extreme position in terms of its search strategy. Most systems use some form of best-first tree search with procedures for backtracking when the analysis requires it. By contrast, the **DRAGON** system uses a complete optimal search, which would be like a breadth-first tree search except the Markov model reduces the tree search to a much smaller network search.

The particular implementation which is discussed in Chapter IV is restricted to a strict left-to-right analysis, and the formulas in Chapters II and III have been expressed in that form. It would be possible to generalize this system to have the analysis proceed from any point in the utterance, but because there is already a complete optimal search, there is no advantage in doing so. It is not necessary to start the analysis at "islands of reliability" because any path which gives the correct interpretation of such an island is eventually considered in the optimal search (unlike a best-first search in which analyzing unreliable data first can cause the correct interpretation of later reliable data never to be considered). Because the computation time is a linear function of the length of the utterance there is no computational advantage in breaking the utterance into several pieces.

The remainder of this thesis is divided into three chapters. Chapter II describes the abstract model which is used in the **DRAGON** system. In the **DRAGON** system each source of knowledge is represented as a probabilistic function of a Markov process([**B8**]). Chapter II presents the general mathematical properties for such systems, but omits the details which are specific to speech recognition. Chapter III presents techniques for representing the knowledge sources necessary for speech recognition. Sometimes several alternative techniques are described for representing a particular source of knowledge. Some of the representation techniques described in Chapter III are used in the simple implementation discussed in Chapter IV. Some of the other techniques have been tested in separate modules but not in a complete recognition system. Some of the techniques have not yet been tested. In particular, no attempt has been made to represent a semantic component or even to obtain a weighted probabilistic grammar. Chapter IV describes a speech recognition system, based on the general model of Chapter II, obtained by implementing some of the representation techniques presented in Chapter III. A summary is presented of recognition results for 102 utterances. The system correctly recognized 49% of the 102 utterances and correctly identified 83% of the 578 words.

INTRODUCTION

The **DRAGON** speech recognition system utilizes the theory of a probabilistic function of a Markov process. In this chapter an introduction is given to the general theory. Chapter III explains how the knowledge sources in a speech recognition system can be represented.

Let $Y(1), Y(2), Y(3), \ldots, Y(T)$ be a sequence of random variables representing the external (acoustic) observations. Let $X(1), X(2), X(3), \ldots, X(T)$ be a sequence of random variables representing the internal states of a stochastic process such that the probability distributions of the Y's depend on the values of the X's, but the X's are not directly observed. As a convenient abbreviation we use a bracket and colon notation to represent sequences. Thus, $Y[1:T]$ represents $Y(1), Y(2), Y(3), \ldots, Y(T)$ and $X[1:T]$ represents $X(1), X(2), X(3), \ldots, X(T)$. Let $y[1:T]$ be the observed sequence of values for the random variables $Y[1:T]$.

GENERAL FORMULATION

We wish to make inferences about the sequence $X[1:T]$ in light of the knowledge of $y[1:T]$. For example, we would like to know the conditional probability $\mathrm{PROB}(\ X(t)=j \mid Y[1:T]=y[1:T]\ )$ for each t and j (the conditional probability of a specific internal state at a specific time, given the entire sequence of external observations). Assuming we have a model for speech production, we can evaluate the *a priori* probability $\mathrm{PROB}(\ X[1:T]\ )$. Assuming a model for the generation of acoustic events associated with a specific sequence of internal states, we can evaluate the conditional probability $\mathrm{PROB}(\ Y[1:T]=y[1:T] \mid X[1:T]=x[1:T]\ )$ (That is, the model yields conditional probabilities of external observations, given the sequence of internal states). Thus we know the conditional probabilities in the generative or synthetic form.

We can compute the desired conditional probabilities using Bayes' formula

$$(1)\ \mathrm{PROB}(\ X(t)=j \mid Y[1:T]=y[1:T]\ )$$

$$= \mathrm{PROB}(\ X(t)=j,\ Y[1:T]=y[1:T]\ )/\mathrm{PROB}(\ Y[1:T]=y[1:T]\ )$$

if we can evaluate the factors on the right hand side. The numerator is given by

(2) PROB(**X**(t)=j, **Y**[1:T]=y[1:T])

$$= \Sigma_{x[1:T],x(t)=j} \text{PROB}(\ \mathbf{X}[1:T]=x[1:T],\ \mathbf{Y}[1:T]=y[1:T]\ )$$

$$= \Sigma_{x[1:T],x(t)=j} \text{PROB}(\ \mathbf{Y}[1:T]=y[1:T]\ |\ \mathbf{X}[1:T]=x[1:T]\ ) \text{PROB}(\ \mathbf{X}[1:T]=x[1:T]\ )$$

where the sum is taken over all possible sequences x[1:T] subject to the restriction x(t)=j. (The joint probability of an internal sequence and an external sequence is the product of the *a priori* probability of the internal sequence and the conditonal probability of the external sequence given by the model. The probability for the event **X**(t)=j is obtained by summing over all internal sequences which meet that restriction.) We can evaluate the *a priori* probability that **Y**[1:T] would be y[1:T] as

(3) PROB(**Y**[1:T]=y[1:T])

$$= \Sigma_{x[1:T]} \text{PROB}(\ \mathbf{Y}[1:T]=y[1:T]\ |\ \mathbf{X}[1:T]=x[1:T]\ ) \text{PROB}(\ \mathbf{X}[1:T]=x[1:T]\ )$$

where the the sum is taken over all possible sequences x[1:T]. (The total probability of an external sequence is the sum of its joint probability with all possible internal sequences.)

Therefore

(4) PROB(**X**(t)=j | **Y**[1:T]=y[1:T])

$$= \text{PROB}(\ \mathbf{X}(t)=j,\ \mathbf{Y}[1:T]=y[1:T]\ )/\text{PROB}(\ \mathbf{Y}[1:T]=y[1:T]\ )$$

$$= \frac{\Sigma_{x[1:T],x(t)=j} \text{PROB}(\ \mathbf{Y}[1:T]=y[1:T]\ |\ \mathbf{X}[1:T]=x[1:T]\ ) \text{PROB}(\ \mathbf{X}[1:T]=x[1:T]\ )}{\Sigma_{x[1:T]} \text{PROB}(\ \mathbf{Y}[1:T]=y[1:T]\ |\ \mathbf{X}[1:T]=x[1:T]\ ) \text{PROB}(\ \mathbf{X}[1:T]=x[1:T]\ )}$$

where the sum in the denominator is taken over all sequences x[1:T] and the sum in the numerator is taken over all such sequences subject to the restriction x(t)=j. (This is the probability of the internal event **X**(t)=j conditional on the observed external sequence, as desired.)

The derivation of equation (4) is just a standard application of Bayes' theorem. It represents a formal inversion of the conditional probabilities from the generative form to the analytic form. (Note: The word "analytic" is used here in a special sense. "Analytic" means "taking apart" as opposed to "synthetic," "generative," or "putting together." In terms of our model, the generative form predicts the observations (Y's) in terms of the internal sequence (X's). The analytic form computes the *a posteriori* probability of the X's conditional on the observed Y's.) The speech-recognition knowledge sources provide the conditional probabilities in a generative form. They must be converted into an analytic form to make inferences about a particular utterance from the observed acoustics. However, the formal inversion formula given in equation (4) is not computationally practical since in general the set of all possible sequences $x[1{:}T]$ is prohibitively large. It is necessary to apply the restrictions of a more specific model to obtain a computationally efficient formula.

MARKOV MODEL

The **DRAGON** speech recognition system assumes that the sequences represent a probabilistic function of a Markov process[**B8**]. Specifically, it is assumed that the conditional probability that $X(t)=j$ given $X(t-1)$ is independent of t and of the values of $X[1{:}t-2]$ and that the conditional probability that $Y(t)=k$ given $X(t)$ and $X(t-1)$ is independent of t and of the values of any of the other X's and Y's. Let $B = \{ b_{i,j,k} \}$ and $A = \{ a_{i,j} \}$ be arrays such that

(5) $\mathrm{PROB}(\, Y(t)=y(t) \mid X[1{:}t]=x[1{:}t],\ Y[1{:}t-1]=y[1{:}t-1]\,)$

$= \mathrm{PROB}(\, Y(t)=y(t) \mid X(t-1)=x(t-1),\ X(t)=x(t)\,)$

$= b_{x(t-1),x(t),y(t)}$

and

(6) $\mathrm{PROB}(\, X(t)=x(t) \mid X[1{:}t-1]=x[1{:}t-1]\,)$

$= \mathrm{PROB}(\, X(t)=x(t) \mid X(t-1)=x(t-1)\,)$

$= a_{x(t-1),x(t)}$

This restriction to a Markov model is the fundamental assumption which allows the **DRAGON** system to be practical. In the Markov model the conditional proabilities depend only on X(t) and

$\mathbf{X}(t-1)$ and not on the entire sequence $\mathbf{X}[1{:}T]$ as in equations (1) to (4). This specialization makes it possible to evaluate the desired conditional probabilities by an indirect but computationally efficient procedure.

The Markov assumption might be paraphrased by saying that the conditional probabilities are independent of context, but such a simple statement would be misleading. Since the state space of the Markov process for our speech recognition application has not yet been formulated, the assumption of the Markov properties should be regarded as a prescription to be followed in the formulation of the state space. Specifically, two situations which differ in "relevant" context must be assigned two separate states in the state space of the random variables $\mathbf{X}[1{:}T]$. Then all "relevant" context is included in the state space description, and the conditional probabilities are indeed independent of further context. The fundamental assumption of the **DRAGON** system is that it is possible to meet this prescription and still have a state space of manageable size.

Under the assumptions of equations (5) and (6) we have

(7) $\mathrm{PROB}(\ \mathbf{X}[1{:}s]=x[1{:}s]\ ) = \mathrm{PROB}(\ \mathbf{X}(1)=x(1)\ )(\Pi_{t=2,s}a_{x(t-1),x(t)})$.

(The *a priori* probability of a given internal state sequence is the product of the transition probabilities for all the transitions in the sequence.) To simplify, add a special extra state to the Markov process: let $x(0)$ be this special state and define $a_{x(0),j} = \mathrm{PROB}(\ \mathbf{X}(1)=j\ )$. Similar conventions are assumed throughout this thesis, unless specifically mentioned otherwise. Then

(8) $\mathrm{PROB}(\ \mathbf{X}[1{:}s]=x[1{:}s]\ ) = \Pi_{t=1,s}a_{x(t-1),x(t)}$

Also

(9) $\mathrm{PROB}(\ \mathbf{Y}[1{:}s]=y[1{:}s]\ |\ \mathbf{X}[1{:}s]=x[1{:}s]\ ) = \Pi_{t=1,s}b_{x(t-1),x(t),y(t)}$

(the model-defined probability of an external sequence, conditional on the internal sequence) where $b_{x(0),j,k}$ is defined appropriately. Combining (8) and (9) yields

(10) $\mathrm{PROB}(\ \mathbf{X}[1{:}s]=x[1{:}s],\ \mathbf{Y}[1{:}s]=y[1{:}s]\ ) = \Pi_{t=1,s}a_{x(t-1),x(t)}b_{x(t-1),x(t),y(t)}$

(the joint probability of an internal sequence and an external sequence as given by the Markov model).

To make possible the efficient computation of the sums in equations (3) and (4), we introduce the probabilities of partial sequences of states and observations ([B8]). Using (2) with t=T=s and using (10), we can set

(11) $\alpha(s,x(s)) = \mathrm{PROB}(\ X(s)=x(s),\ Y[1{:}s]=y[1{:}s]\ )$

$= \Sigma_{x[1:s-1]}\Pi_{t=1,s}a_{x(t-1),x(t)}b_{x(t-1),x(t),y(t)}$

where the sum is over all possible sequences x[1:s−1]. (This is the joint probability of the partial external sequence, up to time s, and the event that the process is in state x(s) at time s.) Let

(12) $\beta(s,x(s)) = \mathrm{PROB}(\ X(s)=x(s),\ Y[s+1{:}T]=y[s+1{:}T]\ )$

$= \Sigma_{x[s+1:T]}\Pi_{t=s+1,T}a_{x(t-1),x(t)}b_{x(t-1),x(t),y(t)}$

where the sum is over all possible sequences x[s+1:T]. (This is the joint probability of the partial external sequence from time s+1 to the end, and the event that the process is in state x(s) at time s.) The benefit of introducing the functions α and β is that the values of $\alpha(s,j)$ for a given s can be computed from the values of $\alpha(s-1,j)$. Similarly, β for a given s can be computed from the values of β for s+1.

RECOGNITION EQUATIONS

In fact

(13) $\alpha(s,j) = \Sigma_i\,\alpha(s-1,i)a_{i,j}b_{i,j,y(s)}$

(because every sequence x[1:s] must have x(s−1)=i for some i)

and

(14) $\beta(s,j) = \Sigma_i\,\beta(s+1,i)a_{j,i}b_{j,i,y(s+1)}$

But $\alpha(T,j) = \mathrm{PROB}(\ X(T)=j,\ Y[1{:}T]=y[1{:}T]\ )$ hence

(15) $\mathrm{PROB}(\ Y[1:T]=y[1:T]\ ) = \Sigma_i\, \alpha(T,j)$.

We can compute the conditional probability distribution for $X(t)$

(16) $\mathrm{PROB}(\ X(t)=j\ |\ Y[1:T]=y[1:T]\ )$

$= \mathrm{PROB}(\ X(t)=j,\ Y[1:T]=y[1:T]\ )/\mathrm{PROB}(\ Y[1:T]=y[1:T]\ )$

$= \alpha(t,j)\beta(t,j)/\Sigma_i\, \alpha(T,i)$.

In speech recognition problems, we usually want to know the particular sequence $x[1:T]$ which maximizes the joint probability $\mathrm{PROB}(\ X[1:T]=x[1:T],\ Y[1:T]=y[1:T]\ )$. Again, the problem can be solved by induction from partial sequences ([B9]). Let

(17) $\gamma(t,j) = \mathrm{Max}_{x[1:t-1]}\mathrm{PROB}(\ X[1:t-1]=x[1:t-1],\ X(t)=j,\ Y[1:t]=y[1:t]\ )$

Then γ may be computed by

(18) $\gamma(t,j) = \mathrm{Max}_i\, \gamma(t-1,i)a_{i,j}b_{i,j,y(t)}$.

Notice that equation (18) is just like equation (13) except that Max has been substituted for Σ. It is convenient to save "back-pointers" while computing γ. Therefore, let $I(t,j)$ be any value of i for which the maximum is achieved in equation (18). Then a sequence $x[1:T]$ for which $\mathrm{PROB}(\ X[1:T]=x[1:T],\ Y[1:T]=y[1:T]\ )$ is maximized is obtained by

(19) $x(T) = j$, where j is any index such that $\gamma(T,j) = \mathrm{Max}_i\gamma(T,i)$

and

(20) $x(t) = I(t+1,x(t+1))$, $t = T-1, T-2, \ldots, 2, 1$.

So far the analysis has assumed that the matrices A and B are fixed and known. However, if A and B are not known but must be estimated, then the α and β computed above may be used to obtain a Bayesian *a posteriori* re-estimation of A and B. The matrix A is re-estimated by

$$(21)\quad \hat{a}_{i,j} = \frac{\Sigma_{t=1,T-1}\mathrm{PROB}(\ X(t)=i,\ X(t+1)=j\ |\ Y[1:T]=y[1:T],\ \{a_{i,j}\},\ \{b_{i,j,k}\}\ )}{\Sigma_{t=1,T-1}\mathrm{PROB}(\ X(t)=i\ |\ Y[1:T]=y[1:T],\ \{a_{i,j}\},\ \{b_{i,j,k}\}\ )}$$

$$= \frac{\Sigma_{t=1,T-1}\alpha(t,i)a_{i,j}b_{i,j,y(t+1)}\beta(t+1,j)}{\Sigma_{t=1,T-1}\alpha(t,i)\beta(t,i)}$$

The matrix B is re-estimated by

$$(22)\quad \hat{b}_{i,j,k} = \frac{\Sigma_{t=1,T-1;\,y(t+1)=k}\mathrm{PROB}(\ X(t)=i,\ X(t+1)=j \mid Y[1:t]=y[1:T],\ \{a_{i,j}\},\ \{b_{i,j,k}\}\ )}{\Sigma_{t=1,T-1}\mathrm{PROB}(\ X(t)=i,\ X(t+1)=j \mid Y[1:T]=y[1:T],\ \{a_{i,j}\},\ \{b_{i,j,k}\}\ )}$$

$$= \frac{\Sigma_{t=1,T-1;\,y(t+1)=k}\alpha(t,i)a_{i,j}b_{i,j,k}\beta(t+1,j)}{\Sigma_{t=1,T-1}\alpha(t,i)a_{i,j}b_{i,j,y(t+1)}\beta(t+1,j)}$$

In fact it can be shown ([B8]) that

$$(23)\quad \mathrm{PROB}(\ Y[1:T]=y[1:T] \mid \{\hat{a}_{i,j}\},\ \{\hat{b}_{i,j,k}\}\ ) \geq \mathrm{PROB}(\ Y[1:T]=y[1:T] \mid \{a_{i,j}\},\ \{b_{i,j,k}\}\ ).$$

Thus, each time the re-estimation equations (21) and (22) are used, new matrices are obtained such that the estimated probability of the observations $Y[1:T]=y[1:T]$ is non-decreasing. Since this estimated probability is a continuous function of the matrix entries (in fact, a polynomial with terms as given by equation (10)), and since the matrix entries are constrained to a compact set (because the entries are non-negative and the row sums are 1), this estimated probability must converge for any sequence of matrices obtained by repeated use of the re-estimation equations. Hence the re-estimation given by equations (21) and (22) may be used repeatedly in an attempt to obtain $\{a_{i,j}\}$ and $\{b_{i,j,k}\}$ which maximize $\mathrm{PROB}(\ Y[1:T]=y[1:T] \mid \{a_{i,j}\},\ \{b_{i,j,k}\}\ )$. Thus we can obtain an approximation to maximum likelihood estimates for $\{a_{i,j}\}$ and $\{b_{i,j,k}\}$.

In re-estimating the matrices A and B, the special structure of the speech recognition problem can be used to good advantage. Although it is convenient to use a single integrated model for the actual analysis and recognition of utterances, the re-estimation of the structural matrices can be performed separately for each of the levels in the hierarchy. Also note that any entry in A or B which is zero remains zero in the re-estimations of equations (21) and (22). Therefore we are able to maintain and utilize the sparseness of these matrices in the re-estimation process.

INTRODUCTION

Each of the knowledge sources in a speech recognition system can be represented in terms of the general model of Chapter II. The total hierarchical system also fits such a model, and it is the total system to which the estimation procedures of Chapter II are applied. This chapter explains the representation of knowledge from each of the sources and their integration into the hierarchy.

REPRESENTATION OF ACOUSTIC-PHONETIC KNOWLEDGE

There are several choices as to how to represent acoustic-phonetic knowledge. A decision must be made whether acoustic observations should be preprocessed by specialized procedures or whether the stochastic model should deal directly with the acoustic parameters. The representation problem is easier assuming specialized preprocessing, so consider this case first.

Assume that at each time t ($1 \leq t \leq T$), an acoustic observation is made. Each such observation consists of a vector of values of a set of acoustic parameters, which in the stochastic model is represented by a vector-valued random variable $\mathbf{Y}(t)$. There is a sequence of phones $\mathbf{P}[1:J]$ which is produced during the time interval $1 \leq t \leq T$. Assume that the phones occupy disjoint segments of time; that is, assume there is a sequence $s_0 < s_1 < s_2 < s_3 < \ldots < s_J$ such that $\mathbf{P}(j)$ lasts from observation $\mathbf{Y}(s_{j-1})$ through observation $\mathbf{Y}(s_j - 1)$. (Set $s_0 = 1$, $s_J = T$.)

Let $p[1:J]$ be the actual sequence of phones in an utterance and let $y[1:T]$ be the actual observed sequence of acoustic parameters. For convenience, also introduce a special initialization phone $p(0)$ which is assigned a special value to allow the initial probabilities to have the same form as the transition probabilities later in the sequence. Since the actual times $s_1, s_2, s_3, \ldots, s_{J-1}$ are not known, it is necessary to associate each arbitrary segment of time with some phone. For each pair of times t_1 and t_2 let $S(t_1,t_2)$ be that value of j for which the expression $(\mathrm{Min}(s_j,t_2)-\mathrm{Max}(s_{j-1},t_1))$ is maximized. (That is, we associate with the pair t_1 and t_2 the index of the phone segment which has the greatest interval in common with the interval from t_1 to t_2.) If $t_2 \leq 1$, then set $S(t_1,t_2) = 0$.

The acoustic preprocessor tries to estimate a phonetic transcription from the acoustics alone. By looking for discontinuities or rapid changes in the acoustic parameters, the preprocessor divides the sequence up into K phone-like segments $Y[1:t_1-1]$, $Y[t_1:t_2-1]$, $Y[t_2:t_3-1]$, ... , $Y[t_{K-1}:t_K-1]$. Then an attempt is made to classify each segment $Y[t_{k-1}:t_k-1]$ using some form of pattern recognition procedure. Let $t_0 < t_1 < t_2 < \ldots < t_K$ be the segment boundary times as decided by the preprocessor and introduce the random variable $D(t)$ which is 1 if there exists a k such that $t_k = t$ and is 0 otherwise. Let $F(k)$ be the label assigned by the preprocessor to the segment $Y[t_{k-1}:t_k-1]$. (For completeness, set $t_k = t_0 = 1$ for $k < 0$, and $t_k = t_K = T$ for $k > K$.)

With some pattern matching procedures it is possible to directly estimate conditional probabilities. When using such a procedure, let

$$(1)\ \ B(p,k) = \mathrm{PROB}(\ Y[t_{k-1}:t_k-1]=y[t_{k-1}:t_k-1]\ |\ P(S(t_{k-1},t_k)=p\ )$$

(the probability that segment k corresponds to phone p as estimated by the pattern matching procedure). On the other hand, the pattern matching procedure might yield only a label $F(k)$ representing a best guess as to the underlying phone. In such a case, it is necessary to estimate the conditional probabilities from statistics of performance of the pattern matcher on hand-labeled data. Let $f[1:K]$ represent the actual sequence of labels generated by the pattern recognizer for the utterance being considered. Then set

$$(2)\ \ B(p,k) = \mathrm{PROB}(\ F(k)=f(k)\ |\ P(S(t_{k-1},t_k))=p\ ).$$

(The probability that segment k corresponds to phone p is estimated as the probability that a segment labeled $f(k)$ corresponds to phone p.) where the conditional probability is estimated by the frequency of such events in a set of training utterances.

In addition to estimating the probability of substitutions or confusions, it is necessary to estimate the probability of the preprocessor producing either too many or too few segments. The probability of such events may be estimated from their frequency of occurrence in a set of training utterances. Let

$$(3)\ \ E(p_1,p_2,n) = \mathrm{PROB}(\ D(t_{k-2})=D(t_{k-1})=D(t_k)=1,\ D[t_{k-2}+1:t_{k-1}-1]=0,\ D[t_{k-1}+1:t_k-1]=0\ |$$
$$P(S(t_{k-2},t_{k-1}))=p_1,\ P(S(t_{k-1},t_k))=p_2,\ S(t_{k-1},t_k)=S(t_{k-2},t_{k-1})+n\ ).$$

(The probability that the segmenter finds one boundary between a segment corresponding to phone p_1 and a segment corresponding to phone p_2, given that the phones are actually n positions apart in the sequence of phones.) If the acoustic preprocessor is reliable, then $E(p_1,p_2,n)$ should be small except for n=1 and should be negligible for n>2. In an implementation of the **DRAGON** system which uses an acoustic preprocessor, it has arbitrarily been assumed that $E(p_1,p_2,n) = 0$ for n>4. Note that $E(p_1,p_2,0)$ is undefined and meaningless unless $p_1 = p_2$.

We can now estimate the conditional probability of the sequence Y[1:T] given the sequence P[1:J].

(4) PROB(Y[1:T]=y[1:T] | P[0:J]=p[0:J])

$$= \Sigma_{n[1:K], z(K)=J} B(p(z(k)),k) E(p(z(k-1)),p(z(k)),n(k)),$$

where $z(k) = \Sigma_{i=1,k} n(i)$ and the sum is taken over all sequences n[1:K] such that z(K) = J. (By convention z(0) = 0.) This equation is a special case of equation (9) of Chapter II.

In order to apply the theory of a probabilistic function of a Markov process, it is necessary to specify the transition probabilities for the phone sequence P[1:J]. It is the task of the other sources of knowledge to specify these probabilities. Phonological rules may be represented either directly or indirectly in the estimates of $E(p_1,p_2,n)$ and B(p,k), but all higher levels of the hierarchy deal only with the sequence P[1:J] and are insulated from the acoustics Y[1:T] or the labels F[1:K].

Even if no special preprocessing is assumed, it is not difficult to represent the acoustic-phonetic knowledge, but there is a penalty of extra computation. Direct estimation of the conditional probability PROB(Y[1:T]=y[1:T] | P[1:J]=p[1:J]) is similar to the problem of machine-aided segmentation and labeling([B2]). Similar algorithms have also been used for word-spotting in continuous speech ([B4], [B11]) and for isolated word recognition ([I1]). The essential idea is an elastic change of the time scale to optimally match a sequence of acoustic observations to a sequence of prototypes.

To relate the phones to the acoustic observations requires knowledge of the acoustic phenomena which are expected with each phone. In line with the probabilistic approach, each phone is assumed to be associated with a stochastic process which produces acoustic parameter values for each instance of the phone. The statistical properties of the stochastic process associated with any particular phone are to be estimated from occurrences of the phone in a set of training utterances which have already been segmented and labeled.

Each acoustic observation is to take a value from a finite set D. Assume that for each phone p there is a positive-integer-valued random variable Z_p and a family of random variables $X_p(1)$, $X_p(2), X_p(3), \ldots, X_p(Z_p)$ with values in D. Let $f_{p,n}$ be the conditional probability function

(5) $f_{p,n}(x(1),x(2),x(3),\ldots,x(n)) = \text{PROB}(\ X_p[1:n]=x[1:n] \mid Z_p=n\ )$.

Let $g_p(n) = \text{PROB}(\ Z_p=n\ )$. The interpretation is that Z_p is the duration of an instance of phone p and $X_p[1:z_p]$ are the acoustic observations made during that instance of p.

Let y[1:T] be the sequence of observations made for the utterance being analyzed. Let p[1:J] be the sequence of phones in the utterance. Let U[1:J] be the sequence of boundary times for the phones. That is, $U(1) < U(2) < U(3) < \ldots < U(J)$ and, for each j, P(j) lasts from observation Y(U(j−1)) to observation Y(U(j)−1). Suppose a set of observations Y[1:T] and times U[1:J] are produced by applying in succession the stochastic processes for each of the phones P(1) through P(J) and concatenating the observations, the individual processes being independent. Then the probability of producing the observed sequence is

(6) $\text{PROB}(\ Y[1:T]=y[1:T], U[1:J]=u[1:J] \mid P[1:k]=p[1:J]\ )$

$$= \Pi_{j=1,J}(f_{p(j),u(j)-u(j-1)}(y[u(j-1):u(j)-1])g_{p(j)}(u(j)-u(j-1))).$$

The segmentation and labeling problem consists of finding the correct set of values for the sequence U[1:J]. Representing the acoustic-phonetic knowledge in a speech recognition system is similar, except the transitions among the phones are determined by probabilities specified by other sources of knowledge rather than being a known sequence.

Note that our model is such that for a given k and u[k:J] we can evaluate

(8) $\mathrm{PROB}(\, Y[u(k):T]=y[u(k):T],\ U[k:J]=u[k:J] \mid P[1:J]=p[1:J] \,)$

$$= \Pi_{j=k+1,J}(f_{p(j),u(j)-u(j-1)}(y[u(j-1):u(j)-1])g_{p(j)}(u(j)-u(j-1)));$$

that is, the probability does not depend on $U[1:k-1]$. The process is an example of a probabilistic function of a Markov process with the vector $(k,U(k))$ being the state variable of the Markov process. The problem of machine-aided labeling can be solved by the techniques of Chapter II.

Introduce the function

(9) $\gamma_1(j,t) = \mathrm{Max}_{u[1:J],u(j)=t}(\, \mathrm{PROB}(\, Y[1:t-1]=y[1-1],\ U[1:j]=u[1:j] \mid P[1:J]=p[1:J] \,)\,)$.

That is, $\gamma_1(j,t)$ is the probability of the best sequence leading up to the state (j,t). The function γ_1 may be calculated according to equation (18) of Chapter II. Thus

(10) $\gamma_1(j,t) = \mathrm{Max}_k(\, \gamma_1(j-1,t-k)f_{p(j),k}(y[t-k:t-1])g_{p(j)}(k) \,)$.

Let $K(j,t)$ be any value of k for which this maximum is achieved. Then after γ_1 and $K(j,t)$ have been calculated for all j and t, the best sequence $u[1:J]$ is obtained by

(11) $u(j) = u(j+1) - K(j+1,u(j+1))$

where $u(J) = T$.

If we are willing to assume that $X_p(1), X_p(2), X_p(3), \ldots, X_p(Z_p)$ are independent and indentically distributed and that

(12) $g_p(n) = (1-a)a^n$, for some a independent of p,

then an even simpler computation is possible. It is not claimed that these additional assumptions are realistic (the acoustic properties of real phones are much more complicated). However, they do produce reasonable results with a great savings in computation.

The extra assumptions allow us to ignore the durations of the phones by factoring out a factor which is the same for all sequences $u[1:J]$, namely the factor $(1-a)^J a^T$. Let's reformulate the Markov process, ignoring duration information. Let the state (j,t) correspond to the event $U(j-1) \le t < U(j)$ with $U(j-1)$ otherwise unrestricted (time t occurs during phone $P(j)$). Let $\gamma_2(j,t)$ be the probability for the best sequence leading up to the state (j,t) and producing the sequence y[1:t]. Then γ_2 may be calculated by

(13) $\gamma_2(j,t) = \mathrm{Max}(\ \gamma_2(j-1,t-1),\ \gamma_2(j,t-1)\ )\mathrm{PROB}(\ X_{p(j)}=y(t)\ )$.

Then the sequence u[1:J] may be calculated by

(14) u(k) = (the greatest integer value of t

such that $t < u(j+1)$ and $\gamma_2(j-1,t-1) > \gamma_2(j,t-1)$).

In machine-aided labeling it is only necessary to consider a single sequence p[1:J]. In a speech recognition problem, we wish to maximize not only over all possible sequences u[1:J] but also over all possible phonetic sequences p[1:J], subject to the transition probabilities determined by the higher levels of the hierarchy. The computation of a function like γ_1 or γ_2 is not performed separately at the acoustic level, but is performed on a Markov process representing the integrated hierarchy.

REPRESENTATION OF LEXICAL KNOWLEDGE AND PHONOLOGICAL RULES

This section discusses the computation of the conditional probability PROB(P[1:J]=p[1:J] | W[1:I]=w[1:I]) where W[1:I] is the sequence of words in the utterance and P[1:J] is the sequence of phones. Each word is represented by an abstract network to which we may apply the re-estimation procedure of equations (21) and (22) of chapter II. The prototype word network consists of several columns of nodes (to simplify the discussion, assume that there are exactly two nodes per column) with each node connected to itself and to every node in its column and in the two following columns. Such a network is shown in Figure 1, where only the arcs leaving from one particular node have been shown.

If each node corresponds to a phone, then an arc which stays in the same column represents insertion of an extra segment At this level we are primarily interested in representing insertions (and other phonological phenomena) made by the speaker, but as already mentioned there is always a choice between representing a given phenomenon at this level (where word-level context

GENERAL WORD PROTOTYPE

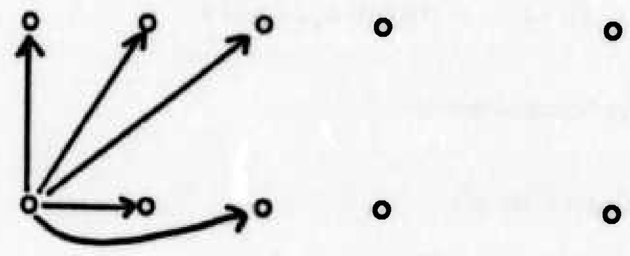


FIGURE 1

is known) or at the acoustic-phonetic level (where only one phone of context is known). An arc which skips a column represents a missed or deleted segment.

Let $Y(t)$ be the phone which occurs at time t. Note that in this hierarchical system, the sequence which is the (unobserved) internal sequence at one level is the external sequence for the next higher level. Whether the acoustic level assumes a preprocessor or not, this next level assumes as its external sequence a sequence of phones (except there are several phenomena which could be represented at either level). Let $X(t) = (X_1(t), X_2(t))$ be the internal state in our abstract word model, where

$1 \leq X_1(t) \leq C$, $X_1(t)$ = column number at time t

$1 \leq X_2(t) \leq R$, $X_2(t)$ = row number at time t

where C is the number of columns in the abstract model and R is the number of rows. For the purpose of this discussion, we take C fixed at the number of phonemes in the canonical version of the word (stored in a dictionary) and take R fixed at 2. Various values of C and R can be used and tested against the actual data.

This abstract network with the associated conditional probabilities represents the probability distribution of possible pronunciations of the word. We assume that the phonetic sequences corresponding to instances of the word are generated by a Markov process. Let

(15) $A((c_1,r_1), (c_2,r_2)) = PROB(X(t)=(c_2,r_2) \mid X(t-1)=(c_1,r_1))$

(16) $B((c,r),p) = PROB(Y(t)=p \mid X(t)=(c,r))$

If we are given a collection of instances of a particular word W, and have estimates for A and B, we can use equations (21) and (22) to re-estimate A and B for the word W. Phonological rules which produce extra segments or deleted segments are represented by A and substitutions are represented by B. Phonological rules which apply across word boundaries can be represented by having several extra states at the beginning and end of each word and having the initial probability distribution depend on the context.

Several variations of this lexical model are also worth considering. If the acoustic level estimates not just the phones but the transemes (pairs of phones as estimated by the acoustic transition between them, as in the ARCS and IBM-Watson systems) then the lexical level should have the distribution of $Y(t)$ depend not just on $X(t)$ but also on $X(t-1)$. It is possible to integrate the acoustic and lexical levels and directly re-estimate the representation of a word in terms of the acoustic parameters. This approach is being followed by Bakis. Another approach is to obtain a network representing the possible pronunciations of a word by applying a list of phonological rules written as production rules and applied to a baseform representation of the word. Automatic procedures for applying such a list of rules for the purpose of speech recognition systems have been developed by Cohen and Mercer[C1] and by Barnett[B5].

The explicit representation of phonological rules in the network is easily achieved at an expense of doubling or tripling the number of nodes in the network. However, it is not essential that an exhaustive set of phonological rules be used. In fact, the implementation of the DRAGON system described in Chapter IV has no explicit phonological rules and only one canonical pronunciation for each word. The reason that this representation is possible is that any phonological phenomena which are not introduced explicitly will be treated at the acoustic-phonetic level. Thus phonological substitutions can be mimicked by adjusting the probabilities in the B and E (equations (1), (2), and (3)) which represent the probabilities of substitutions and insertions and deletions at the acoustic level. The disadvantage of this approach is that the matrices represent less context than is available in the explicit representation of the phonological rules at the lexical level.

There is a serendipitous benefit in using the matrices B and E to represent acoustic-phonetic knowledge independently from the representation of the phonological rules. If the matrices B and E are estimated by running the acoustic preprocessor on a collection of training utterances, then any phonological rules which are left out in the prepared labeling of the training utterances are automatically absorbed into the estimates of B and E. Thus a perfect hand-labeled transcription of the training utterances is not only unnecessary, but undesirable. The best labeling for training purposes is an automatically generated labeling from a procedure knowing the sequence of words and having exactly the same lexical knowledge and phonological rules as the speech recognition system.

REPRESENTATION OF SYNTACTIC AND SEMANTIC KNOWLEDGE

In building the integrated network, the lexical and phonological rule procedures take as input a network representation of the syntax and semantics in which each node of the network represents a word. It is clear that any regular (finite state) grammar can be represented by a finite network. In a speech recognition system the distinction between a regular grammar and an arbitrary context-free or context-dependent grammar is somewhat artificial. Consider the language generated by a particular grammar, not the sequence of words, but the sequence of acoustic events. It is not unreasonable to assume, for example, that the entries in the acoustic-phonetic matrix B(p,k) are all non-zero, although perhaps very small. Such a result would automatically be the case with pattern recognition based on *a posteriori* probabilitities if the conditional probability distributions for the acoustic parameters are multi-variate normal distributions.

But if each entry in B(p,k) is non-zero, then at the acoustic level the language must include all possible sequences. Such a language can, of course, be represented by a finite network grammar. Thus the issue becomes not one of generating the proper language, but rather one of accurately modeling the conditional probabilities. The conditional probabilities may be context-dependent even for a language generated by a context-free grammar. The approach which has been used in the **DRAGON** system has been to enlarge the finite grammar to allow the conditional probabilities to be more accurately represented, but not to try to retain all of the context of the actual language.

The properties of probabilistic grammars have been studied by several investigators ([B10], [E1], [F3], [G2], [H1], [S1], [S2], [T4]). A probabilistic finite state grammar is a special case of a probabilistic function of a Markov process in which the entries in the matrix $\{b_{i,j,k}\}$ of equation (5) of Chapter II are all zeros or ones (only the transitions are probabilistic). Thus such a grammar can be immediately represented in terms of our general model. However, there is still the problem of estimating the transition probabilities.

The general abstract model is not as well suited to representing semantic knowledge as it is to representing the other sources of knowledge which have been discussed. In the implementation described in Chapter IV, there has been no attempt to represent semantic knowledge. In fact, an argument could be made that, since there is no process corresponding to **understanding** the sentence, whatever knowledge is represented by the abstract stochastic model is of necessity not **semantic** knowledge. However, it should be noted that it is not necessary for the stochastic model to directly represent the semantic knowledge itself, but rather it is necessary for the model to represent the influence of the semantic knowledge on the probability distributions of possible sequences of words.

For example, it is possible to have a specialized task-specific module which is capable of understanding the utterances of a given task and which is capable of representing the set of utterances which are possible in a given context. The HEARSAY speech understanding system employs such a mechanism for the VOICE CHESS task. The task is to recognize chess moves that are spoken by a user who is playing a game of chess against the computer. The system has a separate module consisting of a chess playing program, TECH. Not only does the TECH program play chess with the user, but when it is the user's turn to move, TECH lists for the recognition system all moves which are possible in the given position and even rates the moves. Thus the TECH program provides semantic guidance for the recognition system. A similar mechanism may be used to obtain semantic knowledge for the **DRAGON** system. Once the list of legal moves is obtained and rated, this information may be used in setting the transition probabilities for the probabilistic grammar. The fine details may be lost, but much of the information will be represented, the quality of the representation depending on the complexity of the grammar.

There is even a mechanism by which the stochastic model can obtain some semantic information without a specialized module. Consider the goal of mimicking a human being who is trying to guess the next word in an utterance when given some limited amount of context. This person, who is capable of understanding the utterance, could use whatever semantic knowledge is available from the limited context. In this situation the semantic knowledge is more limited than that which is used by the TECH program, which knows the entire sequence of previous moves and hence the current board position, but it is still of value to the speech recognition system. The problem of obtaining the statistics for this type of semantic knowledge is part of the general problem of estimating the transition probabilities for a probabilistic grammar.

The transition probabilities for the grammar network can be estimated from statistics for a set of training sentences. A large set of training sentences should be used, but they only need to be transcribed orthographically, not phonetically, at this level of the hierarchy. If Bayesian statistics are used, the *a priori* probabilities could be set to achieve the same effect as a non-probabilistic use of the grammar. The *a posteriori* probabilities would then be a strict improvement (as judged by performance on the training sentences).

To the extent to which the statistics of the training sentences reflect the true probabilities for spontaneous utterances for the specific task, the probability network represents not only the syntax of the task but also all of the predictive information which can be obtained from the semantics of the available context. That is, if the true probabilities were known, the probability network would be an optimal predictor for a given amount of context, and therefore would predict at least as well as a human who is given the same amount of context and who presumably is capable of understanding the sentence (although the context in this case is not necessarily the whole sentence).

Inter-sentence semantics can also be introduced into the probability network. One way to use inter-sentence semantics is to employ a user model. Suppose there is a model for the user in a particular task such that the the model gives probabilities for the user transitioning among a finite number of states depending on the types of utterances which the user has made. Conceptually this model fits in easily as an extra level of the Markov hierarchy. Computationally it requires that conditional probabilities be estimated separately for each user state. A user model is especially valuable if certain key sentences trigger user transitions with probability one and if for each user state only a small subset of the general grammar is used. Then there is a savings in both the computation and the storage requirements.

SUMMARY

Each of the major sources of knowledge in a speech recognition system can be represented as a stochastic process (usually in more than one way). In speech recognition each knowledge source involves an idealized process **X(1), X(2), X(3), ... , X(T)** which is not observed and a process **Y(1), Y(2), Y(3), ... , Y(T)** depending on the **X** process. The **Y** process is either directly observed or is inferred from lower level knowledge sources in the speech recognition system. Such a dual process can be modeled as a probabilistic function of a Markov process. In the **DRAGON** system such a model is used for each of the knowledge sources.

The speech recognition knowledge sources fit into a hierarchy such that the integrated system also is a probabilistic function of a Markov process. Such a simple general model for speech recognition permits a recognition program which is just a simple implementation of general network search algorithms. Such an implementation of the **DRAGON** system is described in Chapter IV.

INTRODUCTION

In Chapter II, the general properties of a probabilistic function of a Markov process were discussed. Chapter III explained some of the ways in which the knowledge sources of a continuous speech recognition system can be represented by such a model. This chapter describes an implementation of a complete speech recognition system based on these models. This implementation is intended as a preliminary system demonstrating the practicality of building a complete system based entirely on the abstract Markov model. It is not intended as a final system demonstrating the full power of the techniques described here. Each knowledge source is given a simplified representation, and the probabilities in the networks are estimated *a priori* rather than by any automatic re-estimation procedure.

The system is simple, but it is a complete speech recognition system. Starting with knowledge represented in conventional forms—a context-free grammar, a phonetic dictionary, an arbitrary set of acoustic parameters—there is a set of programs for constructing the integrated Markov model, and a general recognition program which can recognize speech for any task based on the integrated network which has been constructed by the other programs. There is some training which is dependent on the talker and on the set of acoustic paramters, but which is independent of the task. This training is done by selecting by hand a set of prototypes for the acoustic segments from a set of utterances by the talker for whom the system is to be trained.

This implementation of the **DRAGON** system consists of five programs: MAKDIC, MAKGRM, MAKNET, GETPRB, and DRAGON. For each program, a brief desciption will be given of what is does and of how it does it. The system has been tested on a set of 102 utterances with about 20 utterances from each of 5 interactive computer tasks. The 5 tasks are VOICE CHESS (the user speaks his moves while playing chess against the computer), DOCTOR (the user asks medical questions and the computer simulates a patient), DESK CALCULATOR (the computer acts as a desk calculator for spoken commands), NEWS (the computer gives the current news stories whose subjects match a spoken specification), and FORMANT (the computer generates various kinds of graphic displays of speech data, according to spoken requests). The grammars for these 5 tasks are given in Appendix B, some sample utterances in Appendix E.

MAKDIC

MAKDIC reads a phonetic dictionary and writes a file describing a network representation for each word in the dictionary. It is this program which would contain any knowledge of within-word phonological rules. Actually, the current implementation of **DRAGON** does not use any explicit phonological rules, so the output of MAKDIC is just a one-to-one translation of the phonetic dictionary. Each word is represented by a linear network with each node connected to itself and to the following node.

A phonetic dictionary including all the words for the 5 tasks is given in Appendix A. The dictionary is written at a very broad phonetic level and has been edited by hand to break up dipthongs and stops into acoustic segments. Certain groups of phones which were distinct in the original dictionary were replaced by a single symbol for each group. This grouping was performed when the phones within a group were practically indistinguishable under the acoustic parameterization used in this implementation. The hand editing was designed to achieve an effect like the lexical model of equations (III.15) and (III.16) of Chapter III, with C=1.

The list of acoustic segment types which appear in the dictionary is given in Table 1. A section of the dictionary is shown in Table 2. The complete dictionary is Appendix A. A flow-chart of the MAKDIC program is shown in Figure 3, and a section of its output file is shown in Table 4. In this implementation, since no phonological rules are applied, the MAKDIC program just goes through the dictionary word-by-word and goes through each word phone-by-phone.

The section of output shown in Table 4 is interpreted as follows: 251 is the index of the word "with" in the dictionary. 4 is the number of phonetic segments in the word. For each of the 4 phonetic segments there are two lines. The first 1 in line 2 is the index of the current phonetic segment within the word. 0 is the internal code for this segment type, "–". The next 1 indicates the number of arcs leading to this node from nodes other than itself. 0 is the probability of this node being skipped. 900 indicates that the probability of the arc from this node to itself is .900. (All probabilities are multiplied by 1000 and truncated to integers.) Next follows a list of all the nodes (other than the node itself) with arcs leading to the current node (in each case there is only one). The 0 in line 3 is the index within the word of the node which has an arc leading to the

ACOUSTIC SEGMENT LABELS

-	silence, pause, voice-bar
AX	(A)BOUT
B	A(B)OUT (release-aspiration portion)
AH	N(U)MBNESS
T	(T)ELL (release-aspiration portion)
AE	H(A)MMING
S	(S)EVEN, (Z)ERO
L	(L)ET
UW	D(O)
F	(F)EVER, WI(TH)
ER	(R)OOK, FEV(ER)
EH	L(E)T
IH	K(I)NG
D	(D)IVIDE (release-aspiration portion)
P	(P)AWN (release-aspiration portion)
N	(N)INE
AO	P(AW)N
AA	(O)CTAL
M	(M)UMPS
SH	BI(SH)OP, MEA(S)URE
K	(K)ING (release-aspiration portion)
IY	QU(EE)N
NX	KI(NG)
G	(G)IVE (release-aspiration portion)
Y	(Y)OU
V	FI(V)E
W	(W)E
OW	ZER(O)
WH	(QU)EEN (release-aspiration and devoiced semi-vowel)
HH	(H)AMMING
UH	R(OO)K

TABLE 1

SECTION OF DICTIONARY

WITH	– W IH F
USING	– Y UW S IH NX
HAMMING	– HH AE M IH NX
HANNING	– HH AE N IH NX
BLACKWELL	– B L AE – K W EH L
RECTANGULAR	– ER EH – K – T EH IH N – G Y UW L AA ER
TRIANGULAR	– T ER AA IH EH IH N – G Y UW L AA ER
FREQUENCY	– F ER IY – K W EH N – S IY
BANDWIDTH	– B AE N – D W IH – D F
CENTER	– S EH N – T ER
CUTOFF	– K AH – T AO F
LOW	– L OW
PASS	– P AE S
HIGH	– HH AA IH

TABLE 2

current node. The 100 indicates that the probability of following this arc is .100. The remaining

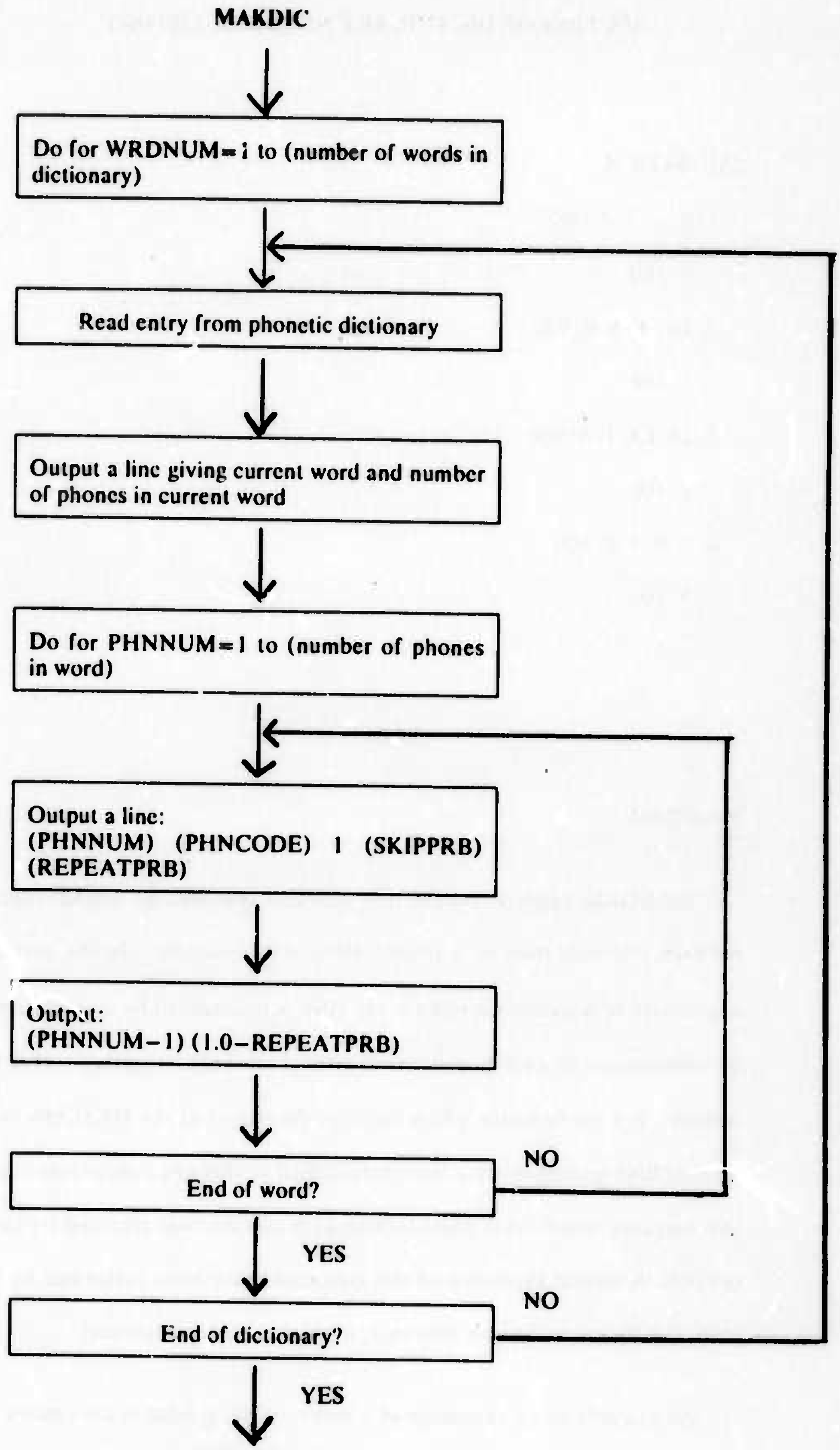


FIGURE 3

phonetic segments are represented similarly.

SECTION OF DICTIONARY NETWORK LISTING

```
251 WITH 4
 1 0 - 1 0 900
   0 100
 2 16 W 1 0 900
   1 100
 3 28 IH 1 0 900
   2 100
 4 7 F 1 0 900
   3 100
```

TABLE 4

MAKGRM

MAKGRM reads a context-free grammar specified by a BNF representation and writes a network representation of a related finite-state grammar. In the current implementation each appearance of a terminal symbol in the BNF is represented by a separate node in the network, but all appearances of each non-terminal symbol are linked together. This linking implies a loss of context. For the tasks for which this implementation of the DRAGON system has been used, the original BNF grammars have been hand edited so that any non-terminal symbol which appeared in two contexts which were important to keep distinct was replaced by two distinct non-terminal symbols. A limited expansion of this type could have been performed by the MAKGRM program itself, but since it was a one-time task, it was done by hand instead.

An example of an expansion of a non-terminal symbol is the symbol <piece> in the VOICE CHESS grammar (Appendix B). The symbol <piece> names the piece taking the action, <pieceb> is part of the location for that piece, <piecec> is a piece being captured, and <pieced>

is either part of the location to which a piece is moving or part of the location on which a piece is being captured.

Note that if either the left contexts or the right contexts are identical f. · two uses of the same non-terminal, then the uses do not need to be distinguished. If the left contexts are identical, then there is no context information to be remembered. If the right contexts are identical, then the left context information does not influence the interpretation of the rest of the sentence. Note that <pieced> has two different uses in the CHESS grammar, with different left contexts, but identical right contexts.

The current version of MAKGRM performs a straight-forward translation of the BNF. Each production is represented by a simple linear network. All the productions with a particular left hand side are linked together with a dummy node at each end. These dummy nodes are then linked to any nodes in the grammar which represent uses of the non-terminal symbol that is the left hand side of these productions. A part of the FORMANT grammar is shown in Figure 5. Figure 6 shows the network in which each production has been represented by a simple linear network. Figure 7 shows the network after the initial and final nodes for each non-terminal symbol have been linked to the uses of that non-terminal. A flowchart for MAKGRM is given in Figure 8.

BNF GRAMMAR

<phr>::=	<spec>
	<phr><spec>
<spec>::=	A <wind> WINDOW OF <num> POINTS
	<num> COEFFICIENTS
	FILE NUMBER <num>
	UTTERANCE NUMBER <num>

FIGURE 5

PARTIALLY CONNECTED NETWORK

<phr>::= <spec>

<phr> ----------------→ <spec>

<spec>::= A ---→ <wind> ---→ WINDOW ---→ OF ---→ <num> ---→ POINTS
<num> ---→ COEFFICIENTS
FILE ---→ NUMBER ---→ <num>
UTTERANCE ---→ NUMBER ---→ <num>

FIGURE 6

SECTION OF GRAMMAR NETWORK

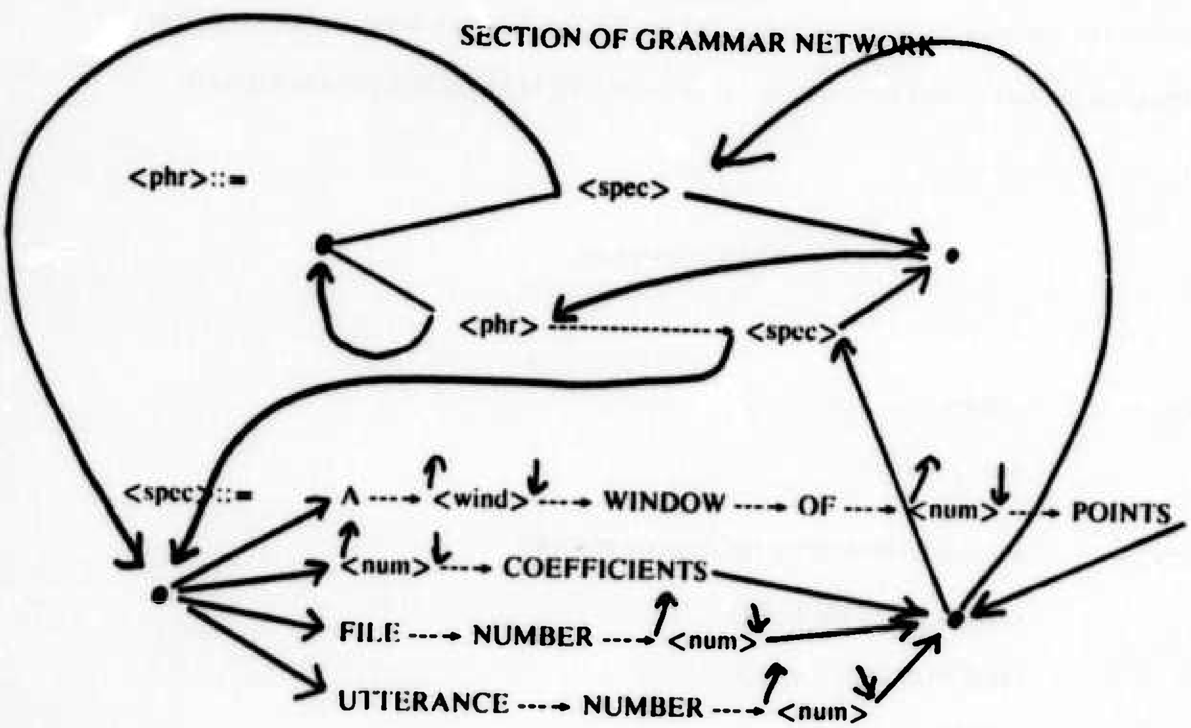


FIGURE 7

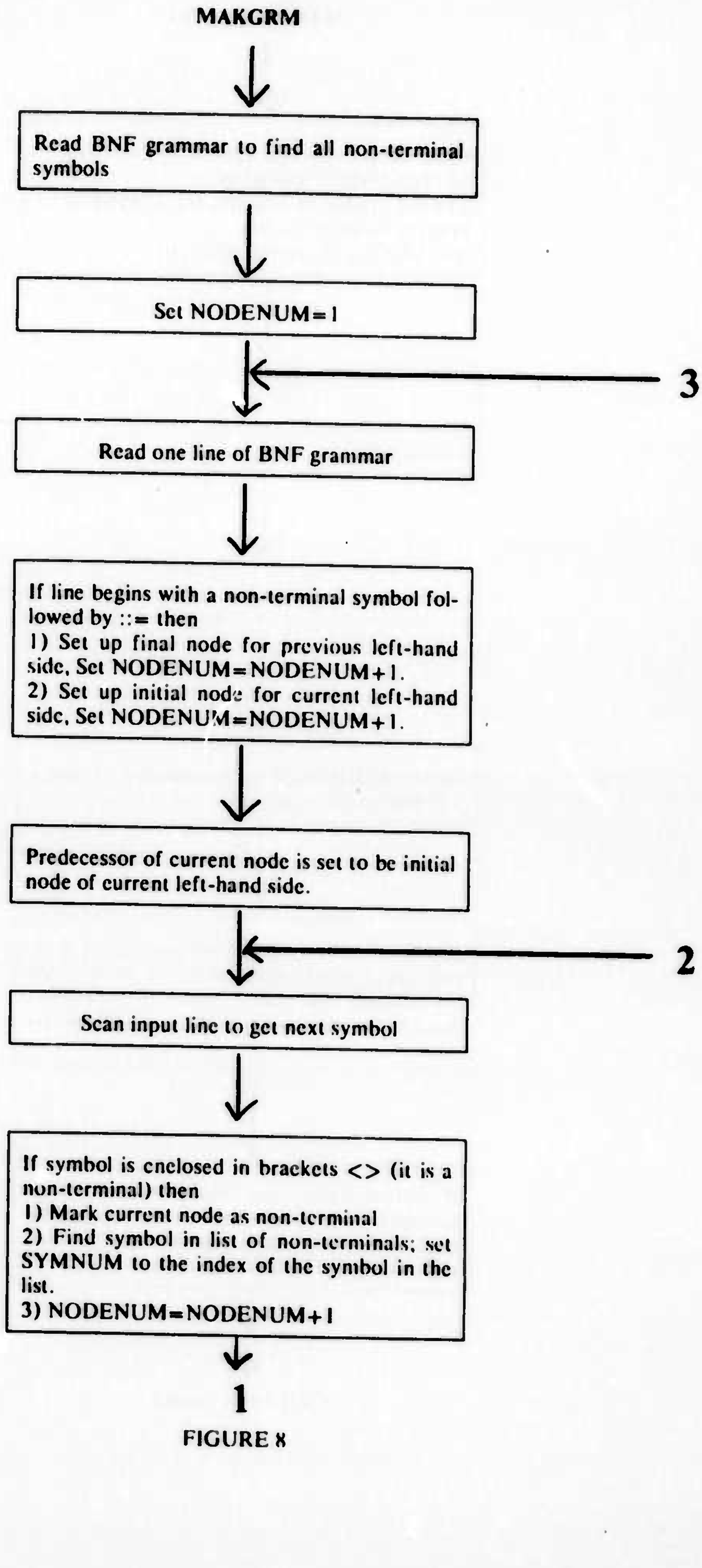

MAKGRM
Read BNF grammar to find all non-terminal symbols
Set NODENUM=1
3
Read one line of BNF grammar
If line begins with a non-terminal symbol followed by ::= then
1) Set up final node for previous left-hand side, Set NODENUM=NODENUM+1.
2) Set up initial node for current left-hand side, Set NODENUM=NODENUM+1.
Predecessor of current node is set to be initial node of current left-hand side.
2
Scan input line to get next symbol
If symbol is enclosed in brackets <> (it is a non-terminal) then
1) Mark current node as non-terminal
2) Find symbol in list of non-terminals; set SYMNUM to the index of the symbol in the list.
3) NODENUM=NODENUM+1
1

FIGURE 8

MAKGRM (cont.)

1

Otherwise symbol is a terminal symbol then
1) Mark node as a terminal.
2) Find symbol in lexicon; set SYMNUM to index of word in lexicon.
3) NODENUM=NODENUM+1

End of line?
If yes then mark last node as the end of a production.

NO → 2

YES

End of grammar?

NO → 3

YES

Do for NODENUM=1 to (number of nodes which have been created)

5

If current node is the initial node for a non-terminal symbol, then introduce an arc into the network connecting each node representing a use of this non-terminal with this initial node.

If current node is the final node for a non-terminal, then introduce an arc connectin each node which ends a production for this non-terminal with this final node.

4

FIGURE 8 (cont.)

MAKGRM (cont.)

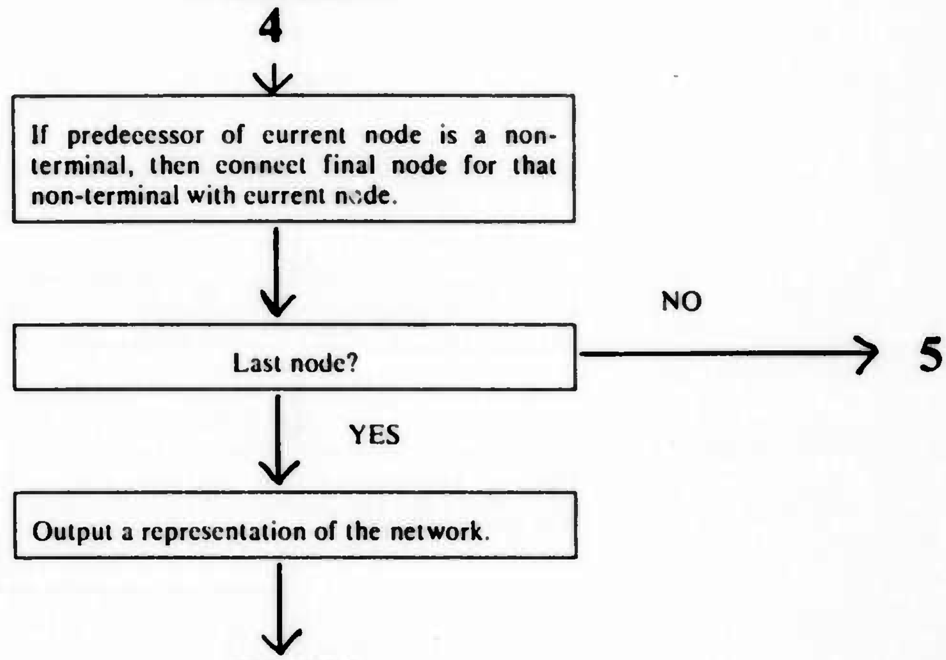


FIGURE 8

MAKNET

MAKNET takes as input a network representation of a grammar (produced by MAKGRM) and a network representation of the dictionary (produced by MAKDIC). It produces an integrated network by substituting the appropriate word network for each node in the grammar network. Phonological rules which apply across word boundaries could be used to adjust the network after the substitution.

MAKDIC, MAKGRM, and MAKNET must keep track of the transition probability associated with each arc of the network. At present simple default values are used. MAKDIC assigns a probability of .9 to any arc leading from a node back to itself, and .1 for any arc leading to the next node. This corresponds to acoustic parameters sampled once every 10 milliseconds, with no presegmentation, and an average phone duration of 100 milliseconds, based on the acoustic-phonetic model of eqations (III.12), (III.13), and (III.14).

The complete input and output for MAKGRM and MAKNET is shown for a simple language in Appendix C. First the simple BNF grammar is given. Next the output file of MAKGRM is shown. Consider the productions with the non-terminal symbol <request> as the left-hand side.

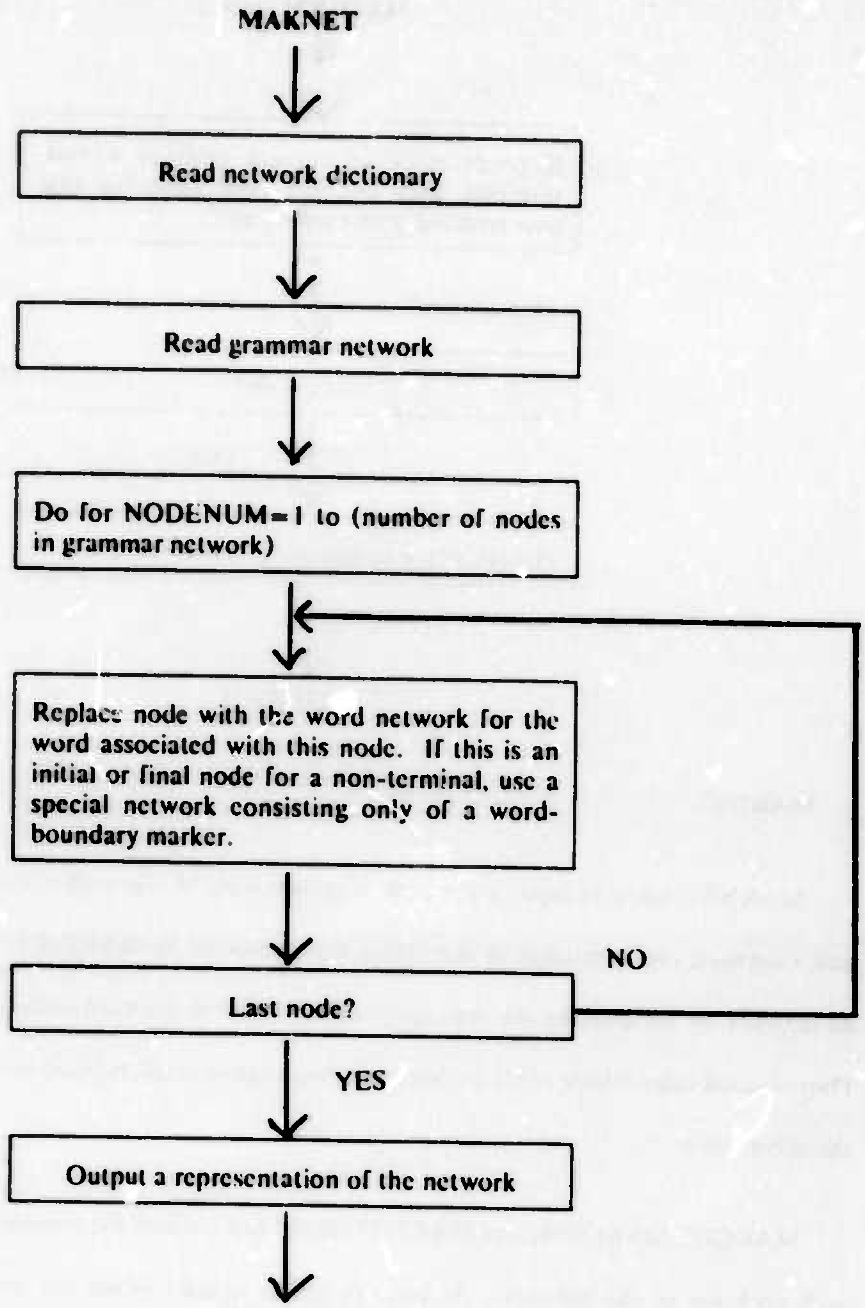


FIGURE 9

The sub-network for these productions begins with the line "<request>::= 6 −2 1." The 6 is the node number for this node, which is the special initial node for this left-hand side. −2 indicates that this node is associated with the second non-terminal symbol. 1 indicates that this node has only 1 arc leading to it. (In this implementation, each arc is listed with the node to which the arc points and transition probabilities are given conditional on the state after the transition, rather than in the conventional form presented in Chapter II. This form has been chosen for the convenience of the implementation, the two theoretical models are equivalent.) 2 (on the next line) is the node number of the node with an arc leading to the current node, and 1000 indicates that the probability of following this arc is 1.000.

"Compute" is the word associated with the next node, which is node 7. It is a terminal symbol and 291 is its index in the dictionary. This node has 1 predecessor, which is node 6 (with probability 1.000). Node 8 is associated with the third (−3) non-terminal symbol <func-phr>. The node has 1 predecessor, node 7. Node 9 is associated with the word "Use" which has index 222. The node has 1 predecessor, node 6 (which is the initial node for this set of productions). Node 10 is associated with the non-terminal symbol <param phr>, and its only predecessor is node 9. Node 11 is the final node for this set of productions (with <request> as the left-hand side). It has two predecessors, node 17 and node 32, which are equally likely. Node 17 is the final node for the productions for the symbol <func-phr>, which is associated with node 8. Node 32 is the final node of the productions for the symbol <param-phr>.

MAKGRM assigns an equal probability to all arcs leading to the same node. This default condition implies that the **DRAGON** system is currently using no semantic knowledge, not even statistically (except for any semantic knowledge which is included in the grammar itself).

The output of MAKNET is a combination of the outputs of MAKDIC and MAKGRM. Each node corresponds to an acoustic segment. Except at word boundaries, each node has only one predecessor besides itself. Notice that there are many nodes marked "−". These silence nodes are common because the dictionary indicates that every word begins with a silence (because the word may be preceded by a pause). The dynamic time warping is sufficiently powerful that these silences can be allowed throughout the network. If no silence is actually present in the acoustic signal, then the dynamic time warping will shrink the duration of time assigned to the "−" node to a single 10 millisecond segment.

GETPRB

GETPRB takes as input a set of acoustic parameter values and produces as output a vector of probability estimates. Each entry in the probability vector represents the conditional probability of producing the given set of acoustic parameter values, conditional on the actual phone at the time of the acoustic observation being the phone corresponding to that particular position in the probability vector.

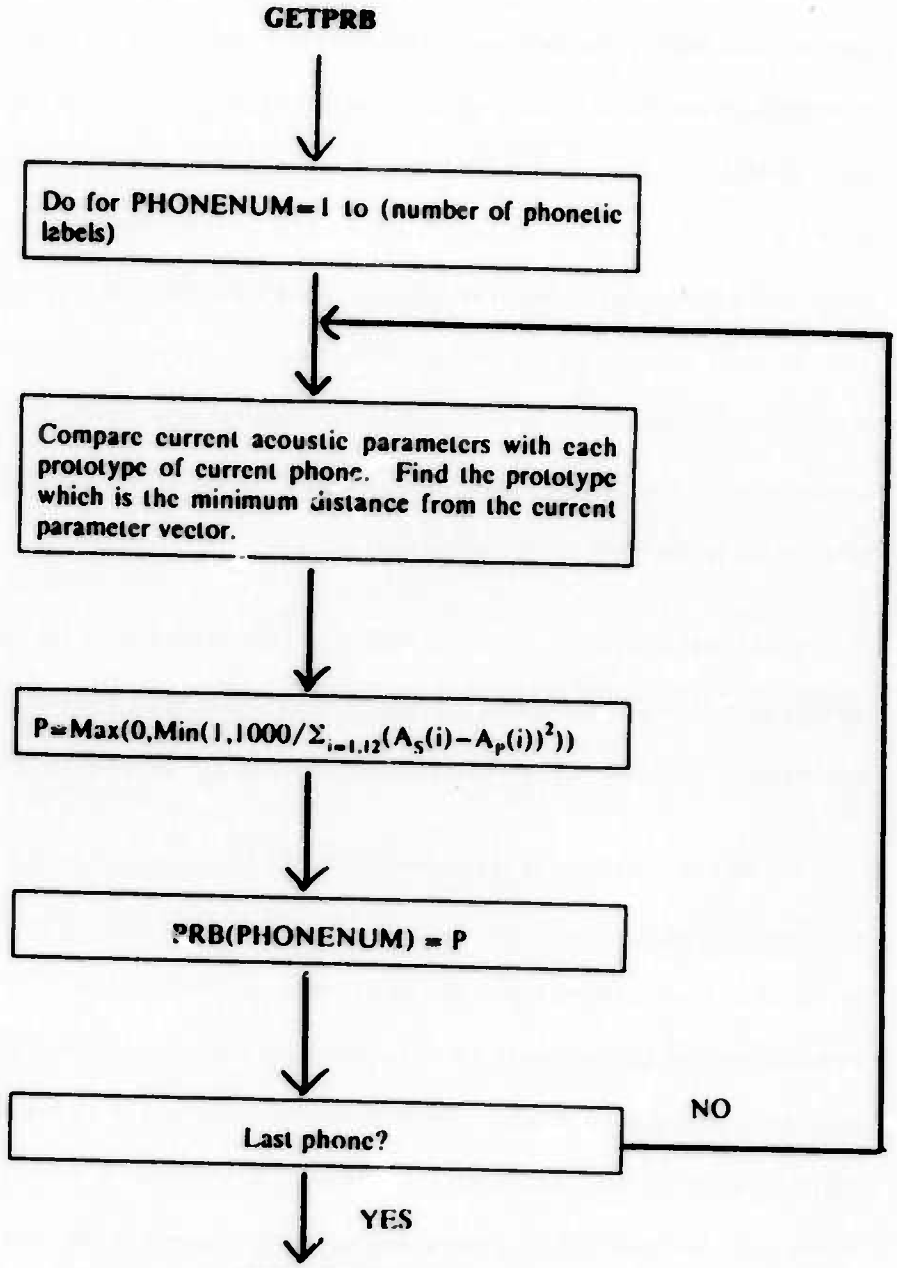


FIGURE 10

Any convenient set of acoustic parameters and any matching procedure could be used here. The current version of the **DRAGON** system uses 12 acoustic parameters sampled once every 10 milliseconds. The basic parameters are an amplitude measure and a zero-crossing-count for each of five filter bands, and for the unfiltered signal. The five filter bands are

A1, Z1: 200-400 Hertz

A2, Z2: 400-800 Hertz

A3, Z3: 800-1600 Hertz

A4, Z4: 1600-3200 Hertz

A5, Z5: 3200-6400 Hertz

AU, ZU are for the unfiltered signal.

The vector of twelve parameters is normalized in a non-linear fashion by dividing A1, Z1, A2, Z2, A3, Z3, A4, Z4, A5, Z5 each by the sum of the twelve paramters and multiplying by 1000. No attempt has been made to find an optimal non-linear transformation; this transformation has been selected by informal experimentation with a small number of alternative transformations. The reason a transformation is introduced is that so many of the consonants are so low in amplitude in **all** the bands that they are difficult to separate by any simple metric. The measurements on the unfiltered signal, AU and ZU, are not normalized, so they retain the information of overall amplitude.

The amplitude measures and zero-crossing counts are normalized together because, especially for the low amplitude cases that we are trying to separate, the zero crossing counts also give a kind of amplitude measure. This phenomenon occurs because the zero crossing counter only counts cycles which exceed a certain threshold. Thus for signals whose amplitude is near the threshold, the zero crossing count is actually a sensitive measure of the amplitude. For strong signals the zero crossing count measures the frequency of the major spectral peak within a particular band.

GETPRB measures the distance between a particular vector of (normalized) acoustic parameter values and a particular prototype by a simple Euclidean distance. However, there are several prototypes for each phone. The prototypes were selected by hand from a set of 50 training sentences spoken by the same talker as the one on whom the system has been tested.

One prototype for each phone was found among the 50 sentences by hand. Each prototype was just the (normalized) vector of acoustic parameter values for some 10 millisecond segment occuring during an instance of the desired phone. Using the GETPRB from these initial prototypes, **DRAGON** was run as a machine-aided labeling program on the same 50 sentences (that is, **DRAGON** was told the sequence of words in each sentence, but not the times at which they occured).

The output of the machine-aided labeling was then carefully checked by hand (there were about one or two corrections per sentence). The labels produced by GETPRB were then compared with this hand-checked segmentation. Whenever there was a steady-state acoustic segment for which no prototype had probability greater than .1, a new prototype was added for the phone which the hand segmentation marked as occuring at that time.

An arbitrary transformation is applied to convert the Euclidean distance measure to an estimate of the conditional probability. The transformation is given by equation (1).

(1) $P = \text{Max}(0, \text{Min}(1, (1000 / (\Sigma_{i=1,12}(A_S(i) - A_P(i))^2))))$,

where $A_S(i)$ is the value of the i th acoustic parameter for the current sample, and $A_P(i)$ is the value of the i th acoustic parameter in the prototype.

A sample of the acoustic labeling produced by GETPRB is given in Appendix D for a portion of the utterance "Use a Hamming window of five hundred twelve points." First a table of the values of the 12 (normalized) acoustic parameters is given; then a table of the top 7 prototypes for each 10 millisecond segment is given. Each row in each table represents one 10 millisecond segment. The segment number is in the first column. In the parameter table the remaining columns are the values of Z1, A1, Z2, A2, Z3, A3, Z4, A4, Z5, A5, ZU, and AU, respectively.

In the table of labels, each label is followed by a number which is its index in the list of prototypes. Frequently several prototypes for the same label occur among the top 7 prototypes. The final two columns are the squares of the Euclidean distances from the current set of acoustic parameter values to the best and second best prototypes.

From time 95 to time 108, the parameters are almost all 0, and "–" is the best prototype. Then "Y" is the best label from 109 to 111. "UW" is best, or one of the best, from 113 to 134. Occasionally another label (IY, AX, L) is rated best, but none of these labels scores high throughout the time from 113 to 134. This section of time would reliably be marked as "UW," from the acoustic information alone. The section from 136 to 138 is a transition between the "UW" and the "S," and no label scores well. From 139 to 144 is the "S." Notice that parameters A4 and Z4 are 0 throughout this segment. This is a feature for distinguishing "S" from "SH," and the system reliably labels "S" and "SH" with these acoustic parameters.

There is no real acoustic evidence for the word "a," and the vowels and nasals of the word "Hamming" are not very clear. At this point the value of an integrated system with other sources of knowledge becomes clear. Rather than doing segmentation and labeling from the acoustics alone, the system makes all decisions in terms of the integrated network representation. The system was able to select, using the labels shown here, the word "Hamming" over all alternatives, including the word "Hanning." However, the system missed the word "twelve" later in the utterance.

DRAGON

The main recognition program, DRAGON, is just an implementation of equations (18), (19), and (20) of Chapter II. The B matrix is provided in implicit form by the procedure GETPRB. The A matrix is represented by the network produced by MAKNET and the default transition probabilities. In comparison with a general transition matrix, the matrix is very sparse (almost all of its entries are zero). The network corresponds to a compacted representation of the transition matrix. Each node in the network corresponds to a row of the matrix, and each non-zero entry in that row corresponds to an arc in the network leaving that node. Since there are usually only two non-zero entries per row, the representation is very compact. Thus the 2356x2356 element transition matrix for the formant tracking task is stored in a few thousand memory locations.

Equation (20) of Chapter II requires that a back pointer be saved telling the best way to get to each node at each point in time. Again it is possible to make use of the extreme sparseness of the A matrix. Since a list is kept of all arcs leading to a given node, a compact back pointer can be kept using only enough bits to select one of the short list of arcs. These back pointers are stored as variable length bytes, fitting as many pointers per memory location as possible. This packed representation of the back pointers makes it possible for the current version of DRAGON to keep

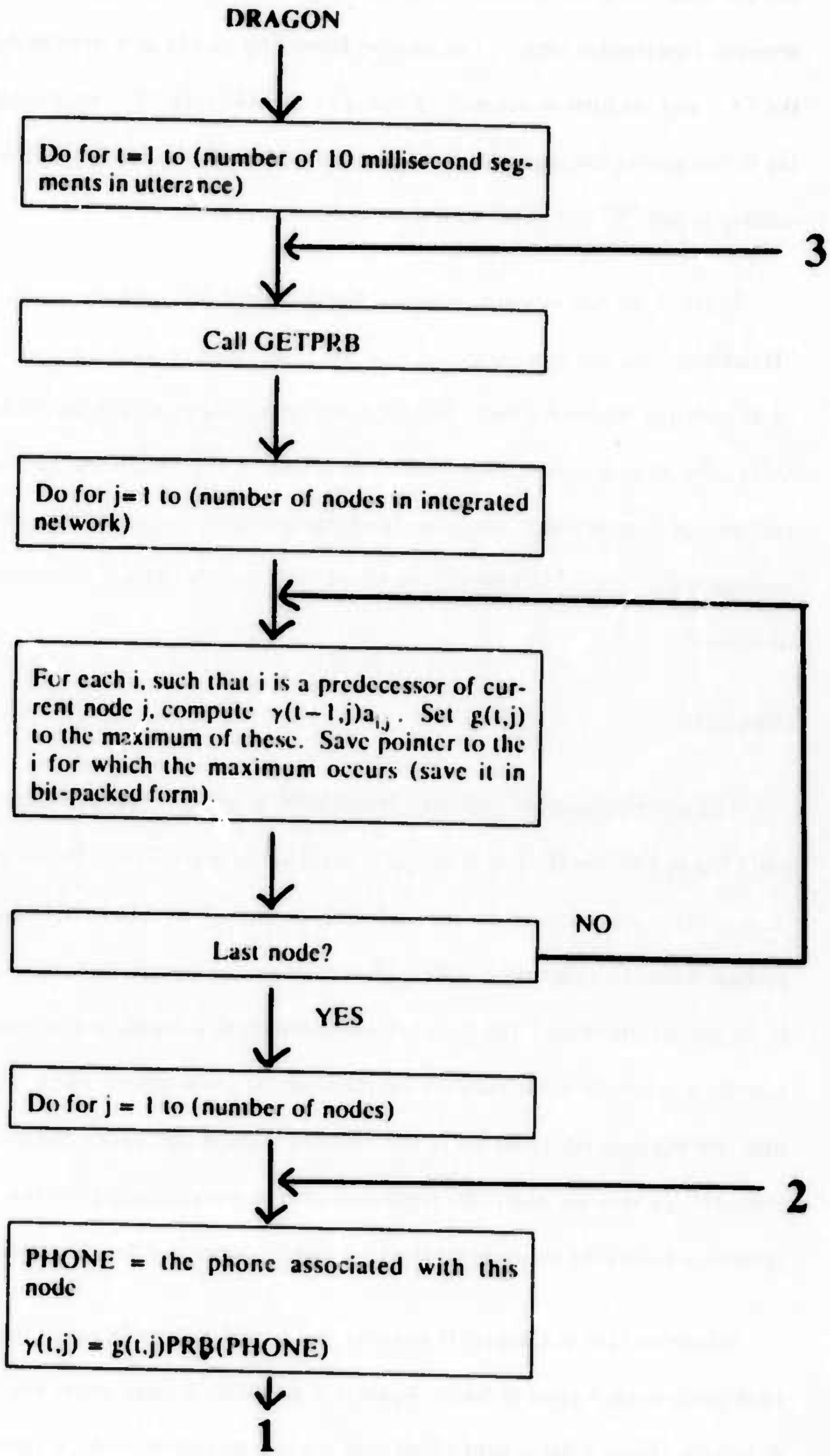


FIGURE 11

all the back pointers for a six second utterance in core memory. In fact, the back pointers for a given 10 millisecond segment for the formant tracking task fit in 73 memory locations (36 bits each).

DRAGON (cont.)

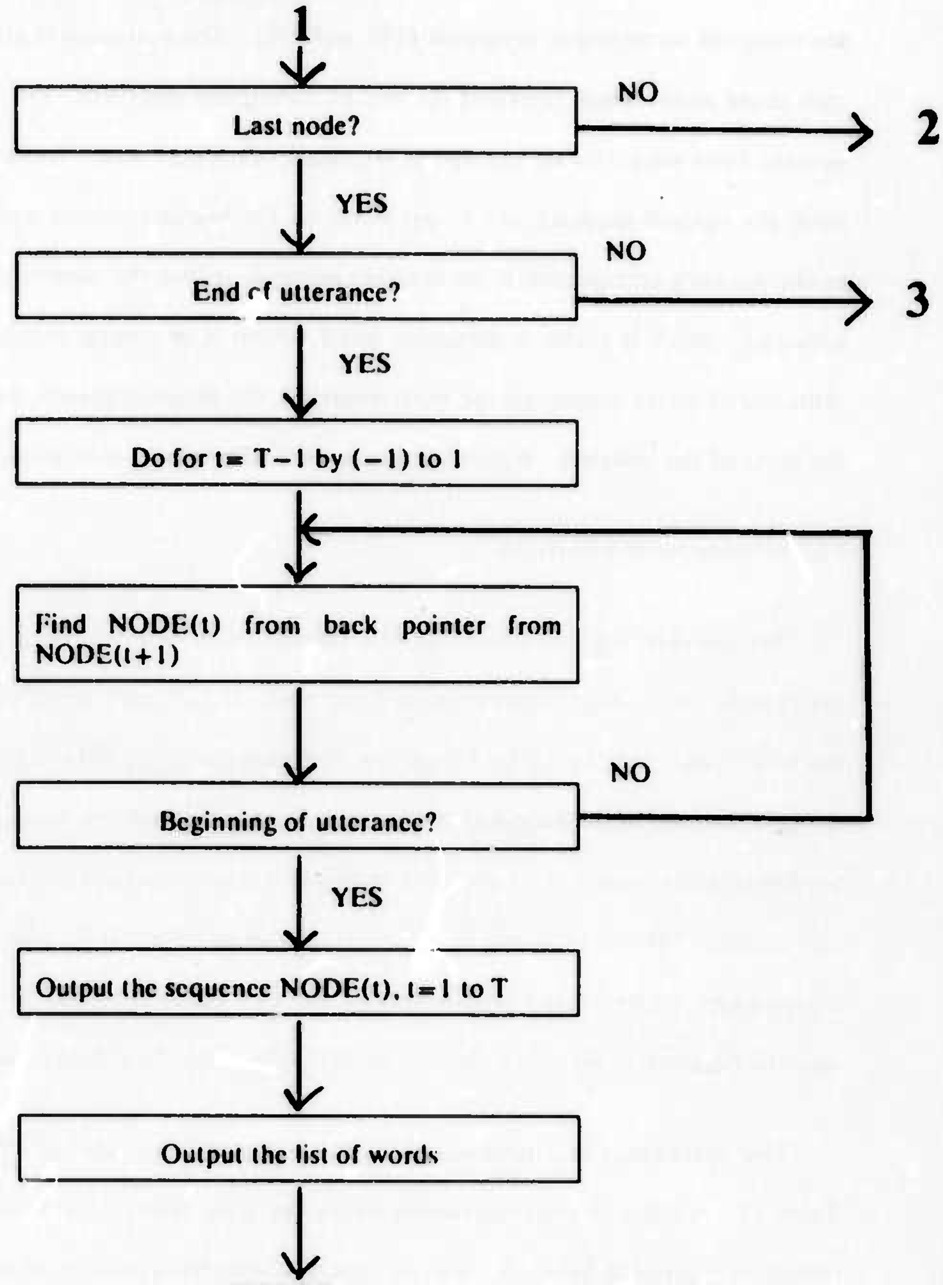


FIGURE 11

A flowchart of the DRAGON program is shown in Figure 11. The program performs the computation of equation (18) for $t = 1, T$. Each node j is considered in turn. Since in this implementation the implicit $b_{i,j,k}$ is independent of i, the value of i for which the maximum occurs in equation (18) depends only on $\gamma(t-1,i)$ and $a_{i,j}$. This value is found and saved as a back pointer. If p is the phone corresponding to node j, then the $b_{i,j,k}$ for the current acoustic parameter values is the number which GETPRB returns in position p of the probability vector. The computation of $\gamma(t,j)$ is completed by multiplying by this factor.

Once the computation of equation (18) has been done for t = 1 through T, the back pointers are retrieved according to equations (19) and (20). The maximum in equation (19) is taken only over those nodes which represent the end of a complete utterance. For the grammars which have actually been used, this set has always consisted of a single node. As the back pointers are traced back, the optimal sequence of internal states for the Markov process is obtained. Since each node in the network corresponds to an acoustic segment within the acoustic realization of a particular phoneme, which is within a particular word, which is in a particular place in the grammar, the sequence of states determines the word sequence, the phone sequence, the segmentation times, and the parse of the sentence. Whichever sequence is of interest can be printed out.

PERFORMANCE RESULTS

The current implementation of the **DRAGON** system has been tested on a total of 102 utterances, with about 20 utterances from each of five interactive computer tasks (described briefly on page 34). In Tables 12-14, the performance of the **DRAGON** system is compared with the performance of the HEARSAY speech understanding system. Because this implementation of the **DRAGON** system has no semantic component, the semantic module of the HEARSAY system was disabled for this experiment. These results were obtained by Lowerre[L3] in a study of the comparative strengths and weaknesses of the two systems. Both of the systems used the 12 acoustic parameters described above, sampled once every 10 milliseconds.

The percentage of utterances correctly recognized in each task by each system is given in Table 12. All 102 of these utterances are by the same talker. The percentage of words correctly identified is given in Table 13. The amount of computation time required by the current system is given in Table 14. These times are the amount of central processor time on a PDP-10 computer as a multiple of the length of the utterance.

Overall the **DRAGON** system recognized 49% of the 102 utterances and identified 83% of the 578 words. An utterance is counted as being correctly recognized if all of the words in the utterance are correctly analyzed. Because of factors such as varying sentence length, the percentage of words correctly identified is more stable for different tasks than the percentage of utterances recognized. Notice that the **DRAGON** system maintained a level of 84% of the words correctly

ACCURACY OF UTTERANCES RECOGNIZED

Task	size of lexicon	no. of utts	Hearsay % correct	Dragon % correct	Hearsay % missed	Dragon % missed
Chess	24	22	32	68	9	0
Doctor	66	21	24	76	33	0
DesCal	37	23	22	17	13	8
News	28	18	50	50	11	0
Formant	194	18	33	33	44	5
		102	31	49	21	3

The % correct figure is the percent of the total utterances that were correctly recognized. The % missed figure is the percent of the total utterances that were completely missed, i.e. no words were correctly identified.

TABLE 12

ACCURACY OF WORDS IDENTIFIED

Task	size of lexicon	no. of words	Hearsay % correct	Dragon % correct
Chess	24	130	69	94
Doctor	66	92	49	88
DesCal	37	116	53	63
News	28	98	74	84
Formant	194	142	33	84
		578	55	83

TABLE 13

identified on the interactive formant tracking task.

The FORMANT task is considerably more complex than the other tasks. It has a vocabulary of 194 words and an infinite language with approximately 16^n sentences of length n words. Each of the other tasks has a finite language with the number of possible sentences ranging up to several hundred million. The HEARSAY system was able to recognize 33% of the utterances for this task, but it only identified 33% of the 142 words. It missed 44% of the utterances completely, and the standard deviation of its computation time is higher than for the other tasks.

This implementation of the DRAGON system was developed using training sentences (by the

TIME NEEDED FOR RECOGNITION

	Hearsay			Dragon			
Task	ave. times real time	Std. Dev.	SD/ave	ave. times real time	Std. Dev.	SD/ave	Size of Dragon network
Chess	13.7	2.6	.19	48.0	.6	.013	410
Doctor	9.4	3.8	.40	67.4	1.1	.016	702
DesCal	15.5	9.4	.61	83.1	1.0	.012	916
News	10.8	6.4	.59	54.7	.6	.011	498
Formant	44.4	23.5	.53	173.8	3.3	.019	2356

For the **DRAGON** system:

(recognition time) = (utt length)(20.9 + .067(net size))

This is accurate to within about 3%.

TABLE 14

same talker) from the tasks CHESS, DOCTOR, and FORMANT. The HEARSAY system was developed for tasks CHESS, DOCTOR, DESCAL, and NEWS. In no instance were any of the utterances used in training the systems included in the test results reported here. One reason the performance of the **DRAGON** system on the DESCAL task was inferior to its performance on the other tasks is that the DESCAL task includes several words which are syntactically equivalent and which are phonetically similar under the analysis used by the current system. No attempt has been made to provide extra phonetic prototypes for this task.

The small standard deviation in processing time for different utterances within a task is a feature of the optimal search algorithm used in the **DRAGON** system. A complete search is done for the globally optimum path through the network. The Markov model allows this global optimum to be found in a time which is proportional to the length of the utterance. If the words are clear and easily recognized, the complete search takes just as long as when the words are unclear and difficult to recognize. On the other hand, the system never takes longer than this fixed time, and it always finds some path through the network. In Table 15, results are given for an earlier version of the **DRAGON** system for each of the 18 utterances in the FORMANT task. The property which should be noticed in these figures is that the processing time does not depend on how many errrors are made in analyzing an utterance.

ACCURACY AND TIME FOR INDIVIDUAL UTTERANCES

Task: Interactive Formant Tracking

Phrase#	#In	#Out	#Cor	#SemCor	Length	Main	Aco
1	6	6	6	6	2170	126.9	18.7
2	9	8	8	8	4270	119.4	18.7
3	8	8	8	8	3730	119.4	18.3
4	9	8	7	7	3690	118.5	18.6
5	7	7	5	5	3490	123.7	18.6
6	9	9	9	9	5670	115.9	18.5
7	10	10	10	10	4510	121.2	18.4
8	7	7	7	7	3200	124.5	18.3
9	11	11	11	11	5120	118.1	17.6
10	7	6	6	6	3300	120.0	17.5
11	4	4	4	4	3070	119.6	18.5
12	10	9	8	8	4480	118.0	18.7
13	4	4	4	4	2760	124.0	18.8
14	4	3	0	0	2300	131.2	18.5
15	10	9	8	9	4260	126.3	19.2
16	11	11	7	8	5160	119.7	18.7
17	10	10	8	9	4060	121.9	17.9
18	6	6	6	6	3110	123.4	17.9

(words correct)/(words in) = .852
(words correct)/(words out) = .890
(words semantically correct)/(words out) = .919

#In = Number of words in actual (input) phrase
#Out = Number of words in output phrase
#Cor = Number of words correctly identified
#SemCor = Number of words semantically correct (error irrelevant to task)
Length = Duration of phrase in milliseconds
Main = (computation time of main recognition routine)/Length
Aco = (computation time of acoustics module)/Length

TABLE 15

The 18 utterances are shown in Table 16. In each pair the actual utterance is given, followed by the utterance which the **DRAGON** system found as the optimal path in its model. The system correctly recognized 8 of the 18 utterances. If we consider "compare" (in sentence 15) to have the same meaning as "look at", and if we consider "compare A and B" to be equivalent to "compare A with B" (in sentence 9), then 10 of the 18 sentences or 55% are semantically correct. A sophishicated semantic component might be able to correct some of the other errors. Appendix E also shows the correct and estimated utterances for the other two tasks for this implementation

Utterances for Interactive Formant Tracking Task

1) I want to do formant tracking.
   I want to do formant tracking.

2) Use a Hamming window of five hundred twelve points.
   Use a Hamming window of five hundred _____ points.

3) Use utterance number six of file number five.
   Use utterance number six of file number five.

4) Increment the window in steps of one hundred points.
   Increment the window in steps of four points.

5) For each window, display the Fourier spectrum.
   For each window, display the formant tracks.

6) Compute the LPC smoothed spectrum using the autocorrelation method.
   Compute the LPC smoothed spectrum using the autocorrelation method.

7) Compute the roots of the inverse filter using Bairstow's method.
   Compute the roots of the inverse filter using Bairstow's method.

8) Display the imaginary part of the roots.
   Display the imaginary part of the roots.

9) I want to compare the autocorrelation method with the covariance method.
   I want to compare the autocorrelation method and the covariance method.

10) Increment the window by one hundred points.
    Increment the window by one ______ points.

11) Display the FFT spectrum.
    Display the FFT spectrum.

12) Use a Hanning window of two hundred fifty-six points.
    Use a Hanning window of two hundred _____ six hertz.

13) Display the FFT spectrum.
    Display the FFT spectrum.

14) Compute the Hilbert transform.
    Use two points.

15) I want to look at image enhancement with different parameters.
    I want to compare image enhancement with different parameters.

16) Display the spectrogram with a pre-emphasis of six decibels per octave.
    Display the spectrogram to a pre-emphasis of six thousand five hertz.

17) Use a ceiling of thirty with a floor of zero.
    Use a ceiling of ten to a floor of zero.

18) For each utterance display the spectrogram.
    For each utterance display the spectrogram.

TABLE 16

of **DRAGON**, and 9 sentences in the AP News task and 8 sentences in the formant task for an earlier version of **DRAGON**.

By considering the specific words which the system identified incorrectly, it is possible to gain some insight about the places at which the model is weakest and/or the task is most difficult. The errors for the **FORMANT** task are given in Table 17.

ERRORS IN FORMANT TASK

	actual phrase	substitution
2)	twelve	______
4)	one hundred	four
5)	Fourier spectrum	formant tracks
9)	with	and
10)	hundred	______
12)	fifty	______
	points	hertz
14)	(entire sentence missed)	
15)	look at	compare
16)	with	to
	decibels per octave	thousand five hertz
17)	thirty with	ten to

TABLE 17

Six of the twelve places at which errors occur involve numbers. It is not surprising that numbers are the greatest point of weakness. In any context in which a number can occur, any number less than one billion is considered grammatical (sometimes including zero). The system has no source of knowledge other than acoustics to select which of the one billion possible numbers was actually spoken. Recognizing a number imbedded in continuous speech from acoustic information alone is a difficult task, and the one-out-of-a-billion selection is usually beyond the ability of this simple general system.

The prepositions and conjunctions are the second greatest source of errors. These function words are usually short and unstressed, so the acoustic information is very unreliable. Previous speech recognition studies ([T3]) have shown that short words are missed more often than long words, and that unstressed function words are missed even more often than other short words. On the other hand, it is often possible to "understand" a sentence as a whole without correctly identifying all the prepositions and conjunctions.

Of the remaining errors, two are caused entirely by a weakness in the model. The original BNF grammar specifies that a "window" length (sentence (12)) be given as a number of "points," and a "pre-emphasis" be specified in "decibels per octave" or "db per octave." In translating the BNF grammar to a finite state grammar, these restrictions were removed. These restrictions could have been retained in the finite state grammar, but only by having a larger state space. Six copies of the number sub-grammar would suffice to distinguish the uses of number with different right contexts ("points", "hertz", <res-unit>, "coeffficients", "per octave", and end-of-phrase). If these two errors were corrected with an expanded grammar, all of the remaining semantically important errors would be numbers, except for sentences (5) and (14).

The current simple implementation of the **DRAGON** system has been designed merely to demonstrate the practicality and power of its general concepts. Clearly many improvements are possible. For example, the acoustic data could be pre-processed and organized into phone-like segments. Then the calculations represented by equations (II.18) and (II.20) would only need to be done for each segment rather than for each 10 millisecond acoustic parameter sample. This reformulation would speed up the calculation in the main recognition program by a factor of about three or four. Especially for larger tasks, substantial savings in computation time can be achieved by employing less than a complete optimal search. A careful study must be done to determine the trade-offs between performance and amount of computation with sub-optimal techniques. More sophisticated models are possible for the knowledge sources, which ought to improve the performance although they would generally increase the amount of computation. A true probabilistic grammar would allow a statistical representation of some semantics as well as a more accurate grammar.

CONCLUSIONS

Let's review the major features of the **DRAGON** speech recognition system and consider how these features influence the performance of this implementation. Some of the features of the **DRAGON** system contribute to its simplicity and ease of implementation, while others give it its power.

(1) Generative form of the model

The fact that the abstract model represents knowledge sources in a generative form made MAKGRM and MAKDIC much simpler to implement. The **DRAGON** network explicitly represents a finite state grammar. Although the underlying stochastic process is assumed to be Markovian, sufficient context is included in the formulation of the state space so that the finite state grammar is represented exactly. It is not necessary to make any compromise to represent the inverse of grammatical productions based on local context. In this regard the **DRAGON** system shares some of the advantages of the top-down recognition systems. On the other hand, the present implementation is limited to a finite state space, so MAKGRM translates any context-free grammar to a related finite state grammar.

(2) Hierarchical arrangement of knowledge sources

The arrangement of the knowledge sources into a conceptual hierarchy simplifies the implementation of the **DRAGON** system by allowing a modularity that separates the details of the representation of the knowledge sources from the recognition program. In this simple implementation this modularity is expressed in the fact that MAKGRM, MAKDIC, MAKNET, GETPRB, and DRAGON are independent programs with well-defined communication. In a more sophisticated implementation the modularity could progress even further and would be even more valuable.

The hierarchical arrangement is also reflected in the sparseness of the transition matrix for the integrated process. This sparseness has played an important role in this implementation of the **DRAGON** system. The explicit network representation allows us to directly access the non-zero entries of the transition matrix, thus avoiding unnecessary computations in the formal equation (II.18). The bit-packed representation of the back pointers allows the entire recognition computation to be performed using core memory.

(3) Integrated network representation

This implementation of the **DRAGON** system integrates the segmentation and labeling into the hierarchy, so the optimal search algorithm performs the segmentation and labeling along with the word identification and parsing. A price is paid in terms of the amount of computation time because the underlying Markov process steps once for every 10 millisecond segment, rather than once for every phone-like segment. However, even this simple implementation can show the advantage of an integrated system compared to a system attempting to make decisions based on any one knowledge source in isolation. The help which the recognition procedure gets from other sources of knowledge allows the segmentation and labeling to be done reliably even with the crude acoustic parameters and simple metric used in GETPRB.

(4) General theoretical framework

The presence of a general theoretical framework greatly simplified the implementation of the **DRAGON** system. It is this feature which has made it possible to construct a complete speech recognition system with limited manpower. It has been necessary to compromise the theoretical framework in a few places (notably the GETPRB procedure and the lexical model), but in general there has been much less special purpose programming than there would have been without the abstract model. The abstract model has been sufficiently flexible that very few compromises have been necessary in deciding what knowledge to represent (with the important exception of semantic knowledge, which has been omitted entirely). The only significant example is that the grammar represented in the network is a finite state grammar rather than a general context-free grammar. This restriction has not been a significant handicap for the 5 tasks which have been implemented so far.

(5) Optimal stochastic search

The optimal search strategy is probably the most unique feature of the **DRAGON** system. It has a significant disadvantage in requiring extra computation. However, the special features of the Markov model allow an optimal search algorithm for which the amount of computation is not nearly as great as might naively be supposed. This implementation of the **DRAGON** system, despite many drawbacks and simplifications, has shown that an optimal search is possible and practical.

The advantages of optimal stochastic search come from avoiding early decisions which might be wrong. By extending all partial paths in parallel we are, in effect, delaying all decisions until all context, past and future, has been considered. The amount of "context" is determined by the formulation of the Markov state space. In the highly stylized grammars used in these interactive computer tasks, the "context" often reaches all the way back to the beginning of the utterance. Thus the optimal search strategy may delay the decision about the first word of the utterance until the effect of this decision on the entire sentence has been considered.

FUTURE WORK

There are many improvements which can be made even within the framework of the current system. The introduction of a sophisticated acoustic preprocessor, while departing from the philosophy of building an entire system from the same abstract model, would result in a significant increase in computational speed. The techniques for using such a preprocessor within the general **DRAGON** system are described in Chepter III (equations (9), (10), and (11)).

The lexical model could be improved either by introducing phonological rules or by using the general lexical model of Chapter III. Either model could be trained using the procedure represent-ed by equations (21) and (22) of Chapter II.

The syntactic-semantic model would be improved by introducing estimates of the conditional probability distributions into the grammar. Given a task with a known grammar, this estimation mainly involves the collection of statistics for a large corpus of utterances from a dialogue in the inter-active computer task. Even for a task with an unspecified grammar, an attempt can be made to approximate the grammar using the re-estimation procedure of equations (21) and (22) of Chapter II.

The assumption of a finite state space (and hence a finite state grammar) is not essential. Markov processes may have infinite state spaces, and much of the theory used here carries through. There are serious problems which must be solved to obtain a practical implementation, but they are not insurmountable. For example, equation (18) of Chapter II can be generalized to apply to an arbitrary context-free grammar, at the expense of making the number of computations proportional to T^3 rather than to T. By segmenting the utterance into syllables, T would be the number of syllables and T^3 might not be too large.

What general implications can be drawn from the results of the **DRAGON** speech recognition system? The **DRAGON** system differs from most other speech recognition systems in three important ways: (1) the use of Markov models, (2) the use of the same abstract model to represent each of the knowledge sources, and (3) the optimal search strategy.

Since the state space can be formulated to include specific context information, the assumption of the Markov property in the models is not so much an assumption as it is a prescription to be followed in the formulation of the state space. The results for this simple implementation demonstrate that this prescription can be followed well enough to get reasonable recognition while keeping the state space of manageable size. However, because the FORMANT task took 173.8 times real time and because the size of the **DRAGON** network grows with the size of the vocabulary, there is a significant area for future research. Techniques need to be developed which can more efficiently represent more complex tasks.

The use of a general abstract model has greatly facilitated the development of the **DRAGON** system and has important implications. Lowerre ([L3]) has been able to analyze the main recognition program to produce an optimized program which produces identical results but is much faster than the original program. Work is being done to adapt the **DRAGON** system to run on a minicomputer. Newell ([N3]) has suggested that the simplicity of the **DRAGON** system would allow it to be used as a "benchmark" system. Any more sophisticated system must justify its greater complexity by recognizing speech either in less time or more accurately than the **DRAGON** system.

A major motivation for constructing the **DRAGON** system has been to demonstrate that speech recognition based on complete optimal search is practical. Clearly, however, a complete search is not the most efficient procedure. The most important area for future research is to develop techniques such that the complete Markov search is an upper bound on the amount of computation, but such that much less computation time is used exploring parallel paths when the correct path is clear.

00100	"AA"	- AA
00200	"AE"	- AE
00300	"AH"	- AH
00400	"AO"	- AO
00500	"AW"	- AA UH
00600	"AY"	- AA IH
00700	"B"	- B IY
00800	"CH"	- SH
00900	"D"	- D IY
01000	"EH"	- EH
01100	"ER"	- ER ER
01200	"EY"	- EH IH
01300	"F"	- EH F
01400	"FILLER"	-
01500	"G"	- G IY
01600	"HH"	- EH IH - SH
01700	"I"	- AA IH
01800	"IH"	- IH
01900	"IY"	- IY
02000	"JH"	- SH
02100	"K"	- K EH IH
02200	"L"	- EH L
02300	"M"	- EH M
02400	"N"	- EH N
02500	"NULL"	-
02600	"NX"	- IH NX
02700	"OW"	- OW
02800	"OY"	- AO IH
02900	"P"	- P IY
03000	"R"	- AA ER
03100	"S"	- EH S
03200	"SH"	- SH
03300	"T"	- T IY
03400	"UH"	- UH
03500	"UW"	- UW
03600	"V"	- V IY
03700	"WH"	- WH
03800	"Y"	- W AA IH
03900	"Z"	- S IY
04000	"ZH"	- SH
04100	'S	- S
04200	A	- AX
04300	ABOUT	- AX - B AH - T
04400	ABOVE	- AX - B AH V
04500	ABSOLUTE	- AE - B S AX L UW - T
04600	ABSOLUTE	- AE - B S OW L UW - T
04700	ACOUSTIC	- AX - K UW S - T IH - K
04800	ADC	- EH IH - D IY S IY
04900	ADD	- AE - D
05000	ADVANCED	- AE - D V AE N - S - T
05100	AFRAID	- AX F ER EH IH - D
05200	AIRPLANE	- EH ER - P L AE IH N
05300	AIRPLANES	- EH ER - P L EH IH N - S
05400	ALL	- AO L
05500	ALPHA	- AH L F AX
05600	AN	- AE N
05700	AN	- AX N
05800	ANALYSIS	- AX N AE L IH S IH S
05900	ANALYZE	- AE N L AA IH S
06000	AND	- AX N - D
06100	ANESTHETIZED	- AX N EH S - T AX S AX - S - D
06200	ANOTHER	- AH N AH F ER
06300	ARE	- AA ER AX
06400	AS	- AE S
06500	ASPIRATED	- AE S - P IH ER EH IH - T EH - D

06600	ASPIRATION	- AE S - P IH ER AA IH SH AX N
06700	ASTHMA	- AE S M AX
06800	AT	- AE - T
06900	ATAL	- AH - T AA L
07000	ATTACHEO	- AA - T AE - SH - T
07100	AUTOCORRELATION	- AO - T OW - K AO ER EH L EH IH SH AX N
07200	AWFUL	- AO F AH L
07300	BABY	- B EH IH - B IY
07400	BACK	- B AE - K
07500	BACKEO	- B AE - K - O
07600	BAO	- B AE - O
07700	BAIRSTOW	- B AE ER S - T OW
07800	BAKER	- B EH IH - K ER
07900	BALL	- B AA L
08000	BALLEO	- B AA L - O
08100	BALLS	- B AA L S
08200	BANOWIOTH	- B AE N - O W IH - O F
08300	BARRED	- B AA ER - O
08400	BECOMES	- B AX - K AH M S
08500	BEEN	- B AX N
08600	BEGINNING	- B IY - G IH N IH NX
08700	BENT	- B EH N - T
08800	BETA	- B EH IH - T AH
08900	BIRO	- B ER - O
09000	BISHOP	- B IH SH AX - P
09100	BISHOP'S	- B IH SH AX - P S
09200	BLACKWELL	- B L AE - K W EH L
09300	BLEEOING	- B L IY - O IH NX
09400	BOTTLE	- B AA - T L
09500	BOUNOARY	- B AE AA N - O ER IY
09600	BOY	- B AO IH
09700	BURST	- B ER S - T
09800	BY	- B AA IH
09900	CALCULATE	- K AE L - K Y UW L EH IH - T
10000	CAPTURES	- K AE - P - SH ER S
10100	CASTLE	- K AE S L
10200	CASTLES	- K AE S L S
10300	CASTRATEO	- K AE S - T ER EH IH - T AX - O
10400	CAT	- K AE - T
10500	CATEGORY	- K AE - T AX - G AO ER IY
10600	CEILING	- S IY L IH NX
10700	CENTER	- S EH N - T ER
10800	CENTISECONOS	- S EH N - T IH S EH - K AX N - O S
10900	CENTRALIZEO	- S EH N T ER L AA IH S - O
11000	CEPSTRAL	- K EH - P S - T ER L
11100	CEPSTRALLY	- K EH - P S - T ER L IY
11200	CEPSTRUM	- K EH - P S - T ER AH M
11300	CHANGE	- SH EH N - G
11400	CHECK	- SH EH - K
11500	CHEST	- SH EH S - T
11600	CHICKEN-POX	- SH IH - K AX N - P AA - K S
11700	CHINA	- SH AA IH N AX
11800	CHURCH	- SH ER - SH
11900	CIGARETTES	- S IH - G ER EH - T S
12000	CIRCUMCISEO	- S AX ER - K AH M S AX - S - O
12100	CLOUDY	- K L AA UW - O IY
12200	CLUSTERING	- K L AH S - T ER IH NX
12300	COEFFICIENTS	- K OW EH F IH SH IH N - T S
12400	COMMA	- K AH M AX
12500	COMPARE	- K AH M - P AE ER
12600	COMPILE	- K AH M - P AA IH L
12700	COMPUTE	- K AH M - P Y UW - T
12800	CONSIDER	- K AH N - S IH - O ER
12900	CONSTRUCTION	- K AX N - S - T ER AH - K SH AX N
13000	CONTINUOUS	- K AX N - T IH N Y UW AX S

Appendix A—PHONETIC DICTIONARY

13100	COVARIANCE	- K OH V AE ER IY AE N - S
13200	CRAMPS	- K ER AE M - P S
13300	CREAM	- K ER IY M
13400	CREF	- K ER EH F
13500	CURSOR	- K ER S ER
13600	CUTOFF	- K AH - T AO F
13700	CYCLES	- S AA IH - K L S
13800	DB	- D IY - B IY
13900	DEAD	- D EH - D
14000	DEBUG	- D IY - B AA - G
14100	DEBUGGING	- D IY - B AX - G IH NX
14200	DECIBELS	- D EH S IH - B EH L S
14300	DECIMAL	- D EH S M L
14400	DELETE	- D AX L IY - T
14500	DELTA	- D EH L - T AH
14600	DENIALIZED	- D EH N - T L AA IH S - D
14700	DEPRESSED	- D IY - P ER EH S - D
14800	DERIVATION	- D AE ER IH V EH IH SH AX N
14900	DESIGNING	- D AX S AA IH N IH NX
15000	DESIRE	- D IH S AA IH ER
15100	DETAIL	- D IY - T EH IH L
15200	DID	- D IH - D
15300	DIFFERENT	- D IH F ER N - T
15400	DIGITAL	- D IH - G IH - T L
15500	DISPLAY	- D AX S - P L EH IH
15600	DIVIDE	- D IH V AA IH - D
15700	DIVIDES	- D IH V AA IH - D S
15800	DIZZINESS	- D IH S IY N AX S
15900	DO	- D UH
16000	DOG	- D AO - G
16100	DOING	- D UH IH NX
16200	DOMAIN	- D OH M EH IH N
16300	DONE	- D AH N
16400	DOUBLE-U	- D AH - B L Y UW
16500	DOWN	- D AA UH N
16600	DRINK	- D ER IH NX - K
16700	DYNAMIC	- D AA IH N AE M IH - K
16800	EACH	- IY - T SH
16900	EASY	- IY S IY
17000	EDITING	- EH - D IH - T IH NX
17100	EIGHT	- EH IH - T
17200	EIGHTEEN	- EH IH - T IY N
17300	EIGHTY	- EH IH - T IY
17400	ELEVATED	- EH L EH V EH IH - T EH - D
17500	ELEVEN	- IY L EH V AX N
17600	EN-PASSENT	- AA N - P AA S AA N
17700	END	- EH N - D
17800	ENHANCEMENT	- AX N HH AE N S - M AX N - T
17900	EPSILON	- EH - P S IH L AA N
18000	ESTIMATION	- EH S - T IH M EH IH SH AX N
18100	EVER	- OH V ER
18200	EXECUTE	- EH - K S AX - K AA UH - T
18300	EXTRA	- EH - K S - T ER AX
18400	FACT	- F AE - K - T
18500	FACTOR	- F AA - K - T AO ER
18600	FANT	- F AA N - T
18700	FAST	- F AE S - T
18800	FATHER	- F AA DH ER
18900	FATHOM	- F AE F AX M
19000	FEATHER	- F EH DH ER
19100	FEATURE	- F IY - T SH ER
19200	FEVER	- F IY V ER
19300	FEVERISH	- F IY V ER IH SH
19400	FFT	- EH F EH F - T IY
19500	FIFTEEN	- F IH F - T IY N

19600	FIFTY	- F IH F - T IY
19700	FILE	- F AA IH L
19800	FILTER	- F IH L - T ER
19900	FILTERED	- F IH L - T ER - D
20000	FINAL	- F AA IH N L
20100	FIND	- F AA IH N - D
20200	FINDING	- F AA IH N - D IH NX
20300	FIRST	- F ER S - T
20400	FIVE	- F AA AX V
20500	FLAP	- F L AE - P
20600	FLOOR	- F L AO ER
20700	FOOL	- F UW L
20800	FOR	- F AO ER
20900	FORMANT	- F AO ER M AE N - T
21000	FOUR	- F AO W ER
21100	FOURIER	- F AO ER IY EH IH
21200	FOURTEEN	- F AO ER - T IY N
21300	FOURTY	- F AO ER - T IY
21400	FRANCE	- F ER AE N - S
21500	FREQUENCY	- F ER IY - K W EH N - S IY
21600	FREQUENTLY	- F ER IY - K W AX N - T L IY
21700	FRICTIONAL	- F ER IH - K SH AX N L
21800	FRONTED	- F ER AH N - T EH - D
21900	FUNCTION	- F AH N - K SH AX N
22000	GAMMA	- G AE M AH
22100	GET	- G EH - T
22200	GETS	- G EH - T S
22300	GIVE	- G IH V
22400	GLOTTAL	- G L AA - T L
22500	GO	- G OW
22600	GOES	- G OW S
22700	GOES-TO	- G OW S - T AX
22800	GOING	- G OW IH NX
22900	GONORRHEA	- G AA N ER IY AX
23000	GRAMMAR	- G ER AE M ER
23100	GRAMMATICAL	- G ER AX M AE - T IH - K L
23200	GRAPHICS	- G ER AE F IH - K S
23300	GRASS	- G ER AE S
23400	HAD	- HH AE - D
23500	HAMMING	- HH AE M IH NX
23600	HANNING	- HH AE N IH NX
23700	HAVE	- HH AE V
23800	HEAD	- HH EH - D
23900	HEADACHES	- HH EH - D IH AX - K S
24000	HEADLINES	- HH EH - D L AA IH N - S
24100	HELLO	- HH EH L OW
24200	HERE	- HH IH ER
24300	HERTZ	- HH ER - T S
24400	HIGH	- HH AA IH
24500	HIJACKING	- HH AA IH - SH AE - K IH NX
24600	HILBERT	- HH IH L - B ER - T
24700	HOSPITALIZED	- HH AA S - P AX L AX S - D
24800	HOW	- HH AA W
24900	HUNDRED	- HH AH N - D ER EH - D
25000	HYPOTHESIS	- HH AA IH - P AA F IH S IH S
25100	I	- AA IH
25200	ICE	- AA IH S
25300	ILL	- IH L
25400	IMAGE	- IH M IH - SH
25500	IMAGINARY	- IH M AE - G IH N AE ER IY
25600	IMMUNIZED	- IH M Y UW W AX S - D
25700	IN	- IH N
25800	INCREMENT	- IH N - K ER AX M EH N - T
25900	INITIAL	- IH N IH SH L
26000	INJURED	- IH N - SH ER - D

26100	INSERT	- IH N - S ER - T
26200	INSTANCE	- IH N - S - T AE N - S
26300	INTERACTIVE	- IH N - T ER AE - K - T IH V
26400	INTO	- IH N - T UW
26500	INVERSE	- IH N V ER S
26600	IS	- AX S
26700	ISRAEL	- IH S ER IY L
26800	IT	- IH - T
26900	ITAKURA	- IH - T AH - K ER AH
27000	JAMES	- SH EH IH M S
27100	JUDGE	- SH AH - D - SH
27200	KING	- K IH NX
27300	KING'S	- K IH NX S
27400	KNIGHT	- N AA IH - T
27500	KNIGHT'S	- N AA IH - T S
27600	LABEL	- L EH IH - B L
27700	LABELING	- L EH IH - B L IH NX
27800	LABELS	- L EH IH - B L S
27900	LARYNGEALIZED	- L AA ER IH N - G L AA IH S - D
28000	LEARN	- L ER N
28100	LEFT	- L EH F - T
28200	LENGTH	- L AX NX - F
28300	LESION	- L IY S AX N
28400	LESIONS	- L IY S AX N - S
28500	LET	- L EH - T
28600	LILY	- L IH L IY
28700	LINEAR	- L IH N IY ER
28800	LION	- L AA IH UH N
28900	LIP	- EH L AA IH - P IY
29000	LIST	- L IH S - T
29100	LITERAL	- L IH - T ER L
29200	LOAD	- L OW - D
29300	LOCALIZED	- L OW - K L AA IH S - D
29400	LOG	- L AO - G
29500	LOGARITHM	- L AO - G AE ER IH F M
29600	LONG	- L AO NX
29700	LOOK	- L UH - K
29800	LOW	- L OW
29900	LOWERED	- L OW ER - D
30000	LPC	- EH L - P IY S IY
30100	MARKEL	- M AA ER - K L
30200	MARKING	- M AA ER - K IH NX
30300	MATE	- M EH IH - T
30400	MAX	- M AE - K S
30500	MAY	- M EH IH
30600	ME	- M IY
30700	MEASLES	- M IY S L S
30800	MEASURE	- M EH SH ER
30900	METHOD	- M EH F AH - D
31000	METHODS	- M EH F AH - D S
31100	MICROSECONDS	- M AA IH - K ER OW S EH - K AX N - D S
31200	MILD	- M AA IH L - D
31300	MILLION	- M IH L IH AX N
31400	MILLISECONDS	- M IH L IH S EH - K AX N - D S
31500	MIN	- M IH N
31600	MINUS	- M AA IH N AH S
31700	MOD	- M AH - D
31800	MODIFIER	- M AA - D IH F AA IH ER
31900	MOM	- M AA M
32000	MOVE	- M UH V
32100	MOVES	- M UH V S
32200	MOVES-TO	- M UH V S - T AX
32300	MUCH	- M AA - SH
32400	MUMPS	- M AX M - P S
32500	MURDER	- M ER - D ER

32600	NASALIZED	- N EH IH S L AA IH S - D
32700	NAUSEA	- N AO AH SH AX
32800	NEGAT	- N AX - G EH IH - T
32900	NETWORK	- N EH - T W ER - K
33000	NEW	- N UW
33100	NEWTON	- N UW - T AX N
33200	NINE	- N AA IH N
33300	NINETEEN	- N AA IH N - T IY N
33400	NINETY	- N AA IH N - T IY
33500	NIXON	- N IH - K S AX N
33600	NOBODY	- N OW - B AH - D IY
33700	NON-SPEECH	- N AA N - S - P IY - SH
33800	NOW	- N AA UW
33900	NUMBER	- N AH M - B ER
34000	NUMBNESS	- N AH AX M N AX S
34100	NUTS	- N AX - T S
34200	OBOE	- OW - B OW
34300	OCTAL	- AA - K - T L
34400	OCTAVE	- AA - K - T EH V
34500	OF	- AO V
34600	OF	- AX V
34700	OFTEN	- AO AH F AX N
34800	ON	- AO N
34900	ONE	- W AH N
35000	OPERATION	- AH - P ER AE IY SH AX N
35100	OR	- AO ER
35200	ORDER	- AO ER - D ER
35300	OVEREAT	- OW V ER IY - T
35400	PAIN	- P AX IH N
35500	PAINS	- P AX IH N S
35600	PALATALIZED	- P AE L AE - T L AA IH S - D
35700	PARAMETER	- P AX ER AE M EH - T ER
35800	PARAMETERS	- P ER AE M AX - T ER S
35900	PART	- P AA ER - T
36000	PASS	- P AE S
36100	PAWN	- P AO N
36200	PEAK	- P IY - K
36300	PEAKS	- P IY - K S
36400	PER	- P ER
36500	PERIOD	- P IH ER IY AX - D
36600	PHONE	- F OW N
36700	PHONEME	- F OW N IY M
36800	PHONEMIC	- F AX N IY M IH - K
36900	PHONETIC	- F AX N EH - T IH - K
37000	PHRASE	- F ER EH IH S
37100	PICKING	- P IH - K IH NX
37200	PITCH	- P IH - T SH
37300	PLOT	- P L AA - T
37400	PLUS	- P L AH S
37500	POINTS	- P AO IH N - T S
37600	POP	- P AA - P
37700	POSITION	- P AX S IH SH AX N
37800	POSITIONS	- P AX S IH SH AX N - S
37900	POST-EMPHASIS	- P OW S - T EH M F AH S IH S
38000	POT	- P AA - T
38100	POWER	- P AA W ER
38200	PRE-EMPHASIS	- P ER IY EH M F AH S IH S
38300	PREDICTION	- P ER IY - D IH - K SH AX N
38400	PREDICTIVE	- P ER AX - D IH - K - T IH V
38500	PRESENT	- P ER EH S EH N - T
38600	PRIMARY	- P ER AA IH M EH ER IY
38700	PRONY	- P ER OW N IY
38800	PROTOCOL	- P ER OW - T OW - K AO L
38900	PUP	- P AH - P
39000	PUT	- P UH - T

39100	Q	- K AA UH
39200	QUEEN	- WH IY N
39300	QUEEN'S	- WH IY N - S
39400	RABINER	- ER AH - B IH N ER
39500	RAISED	- ER EH IH S - D
39600	RAPE	- ER AE IH - P
39700	RATING	- ER EH IH - T IH NX
39800	REAL	- ER IY L
39900	RECTANGULAR	- ER EH - K - T EH IH N - G Y UW L AA ER
40000	REDUCED	- ER IH - D UW S - T
40100	RELEASED	- ER IH L IY S - T
40200	REQUEST	- ER IY - K W IH S - T
40300	RESOLUTION	- ER EH S OW L UW SH AX N
40400	RETRACTED	- ER IY - T ER AE - K - T EH - D
40500	RETROFLEXED	- ER EH T ER OW F L EH - K S - D
40600	RIGHT	- ER AA IH - T
40700	ROAR	- ER OW ER
40800	ROBINSON	- ER AA - B IH N - S AH N
40900	ROOK	- ER UH - K
41000	ROOK'S	- ER UH - K S
41100	ROOT	- ER UW - T
41200	ROOTS	- ER UW - T S
41300	ROSES	- ER OW S IH S
41400	ROUNDED	- ER AA UH N - D EH - D
41500	RUSSIA	- ER AX SH AX
41600	SAY	- S EH IH
41700	SCALE	- S - K EH IH L
41800	SCHAFFER	- SH EH IH F ER
41900	SCHWA	- SH W AA
42000	SECOND	- S EH - K AH N - D
42100	SECONDARY	- S EH - K AH N - D EH ER IY
42200	SECTION	- S EH - K SH AX N
42300	SEE	- S IY
42400	SEGMENT	- S EH - G M AX N - T
42500	SEGUE	- S EH - G W EH IH
42600	SENTENCE	- S EH N - T EH N - S
42700	SERIOUS	- S IH ER IY AX S
42800	SEVEN	- S EH V AX N
42900	SEVEN	- S EH V EH N
43000	SEVENTEEN	- S EH V EH N - T IY N
43100	SEVENTY	- S EH V EH N - T IY
43200	SEVERE	- S AX V IH ER
43300	SEX	- S EH - K S
43400	SHARP	- SH AH ER - P
43500	SHORT	- SH AO ER - T
43600	SHOULD	- SH UH - D
43700	SHOW	- SH OW
43800	SICK	- S IH - K
43900	SIDE	- S AA IH - D
44000	SILENCE	- S AA IH L EH N - S
44100	SIMULATION	- S IH M Y UW L EH IH SH AX N
44200	SING	- S IH NX
44300	SISTER	- S IH S - T ER
44400	SIT	- S IH - T
44500	SIX	- S IH - K S
44600	SIXTEEN	- S IH - K S - T IY N
44700	SIXTY	- S IH - K S - T IY
44800	SLASH	- S L AE SH
44900	SMOKE	- S M OW - K
45000	SMOOTHED	- S M UW F - D
45100	SMOOTHING	- S M UW F IH NX
45200	SPEAKER	- S - P IY - K ER
45300	SPECIFICATION	- S - P EH S IH F IH - K EH IH SH AX N
45400	SPECTRAL	- S - P EH - K - T ER L
45500	SPECTROGRAM	- S - P EH - K - T ER OW - G ER AE M

45600	SPECTRUM	- S - P EH - K - T ER AX M
45700	SPEECH	- S - P IY - T SH
45800	START	- S - T AA ER - T
45900	STARTING	- S - T AA ER - T IH NX
46000	STATE	- S - T EH IH - T
46100	STEADY	- S - T EH - D IY
46200	STEPS	- S - T EH - P S
46300	STOP	- S - T AA - P
46400	STORE	- S - T AO ER
46500	STORIES	- S - T AO ER IY S
46600	STRESS	- S - T ER EH S
46700	SUB-PHONETIC	- S AH - B F AX N EH - T IH - K
46800	SUB-SEGMENT	- S AH - B S EH - G M EH N - T
46900	SUDDEN	- S AH - D AX N
47000	SUMMARY	- S AX M ER IY
47100	SURGERY	- S ER - SH ER IY
47200	SYLLABIC	- S IH L AE - B IH - K
47300	SYMBOL	- S IH M - B AO L
47400	SYNTHESIS	- S IH N F AX S IH S
47500	TAKE	- T EH IH - K
47600	TAKES	- T EH IH - K S
47700	TASK	- T AE S - K
47800	TELL	- T EH L
47900	TEN	- T EH N
48000	TERTIARY	- T ER SH IY EH ER IY
48100	TESTING	- T EH S - T IH NX
48200	THAT	- DH AE - T
48300	THE	- DH AX
48400	THETA	- F EH IH - T AX
48500	THIN	- F IH N
48600	THIRD	- F ER - D
48700	THIRTEEN	- F ER - T IY N
48800	THIRTY	- F ER - T IY
48900	THORN	- F AO ER N
49000	THOUSAND	- F OW S AE N - D
49100	THREE	- F ER IY
49200	TIME	- T AA IH M
49300	TIMES	- T AA IH M S
49400	TITLE	- T AA IH - T L
49500	TO	- T AX
49600	TRACKING	- T ER AE - K IH NX
49700	TRACKS	- T ER AE - K S
49800	TRAIN	- T ER EH IH N
49900	TRANSCRIPTION	- T ER AE N - S - K ER IH - P SH AX N
50000	TRANSFORM	- T ER AE N - S F AO ER M
50100	TRANSITION	- T ER AE N - S IH SH AX N
50200	TRIANGULAR	- T ER AA IH EH IH N - G Y UW L AA ER
50300	TRILLED	- T ER IH L - D
50400	TUBERCULOSIS	- T UW - B ER - K Y UW L OW S AX S
50500	TWELVE	- T W EH L V
50600	TWENTY	- T W EH N - T IY
50700	TWO	- T UW
50800	TWO	- T W UW
50900	UN-STRESSED	- AH N - S - T ER EH S - D
51000	UNROUNDED	- AH N ER AA UH N - D EH - D
51100	UNTIL	- AX N - T IH L
51200	URINE	- Y ER AX N
51300	US	- AH S
51400	USE	- Y UW S
51500	USING	- Y UW S IH NX
51600	UTTERANCE	- AH - ER EH N - S
51700	VALUE	- V AE L Y UW
51800	VEAL	- V IY L
51900	VELARIZED	- V IY L AA ER AA IH S - D
52000	VIETNAM	- V IH EH - T N AE M

52100	VOICED	- V AO IH S - D
52200	VOICELESS	- V AO IH S L EH S
52300	W	- D AA - B L AA UH
52400	WAGON	- W AE - G AX N
52500	WANT	- W AA N - T
52600	WAR	- W AO ER
52700	WATERGATE	- W AO - T ER - G AE IH - T
52800	WAVEFORM	- W EH IH V F AO ER M
52900	WE	- W IY
53000	WEIGH	- W AO AX
53100	WERE	- W ER
53200	WHAT	- W AH - T
53300	WHEN	- W AX N
53400	WHERE	- W AE ER
53500	WHICH	- WH IH - SH
53600	WINDOW	- W IH N - D OW
53700	WITH	- W IH F
53800	WORD	- W ER - D
53900	X	- EH - K S
54000	Y	- W AA IH
54100	YELLOW	- Y EH L OW
54200	YES	- Y EH S
54300	YOU	- Y AX
54400	YOUR	- Y ER
54500	Z	- S IY
54600	ZERO	- S IH ER OW
54700	ZOO	- S UW
54800	[	-
54900	]	-

```
; SUB-GRAMMAR FOR FORMANT TRACKING SUB-TASK.

<f!form-sent>::=        [ <f!request> ]

<f!request>::=          <f!desire-sent>
                        <f!param-sent>

<f!desire-sent>::=      I WANT TO DU <f!task>

<f!task>::=             FORMANT TRACKING
                        TIME DOMAIN ANALYSIS
                        PITCH MARKING
                        PHONETIC BOUNDARY MARKING
                        PHONETIC LABELING
                        PHONETIC TRANSCRIPTION
                        ACOUSTIC FEATURE LABELING
                        GRAMMATICAL CATEGORY DERIVATION
                        GRAMMAR SPECIFICATION
                        NETWORK EDITING
                        PARAMETER TESTING
                        DEBUGGING
                        SIMULATION
                        HYPOTHESIS RATING
                        FACTOR ANALYSIS
                        CLUSTERING
                        DISPLAY CONSTRUCTION
                        SPEECH SYNTHESIS
                        DIGITAL FILTER DESIGNING

<f!param-sent>::=       <f!command>
                        <f!intro><f!command>

<f!command>::=          USE <f!param-phr>
                        <f!compute><f!func-phr>
                        <f!compute><f!func-phr> USING <f!meth-type> METHOD
                        <f!plot><f!plot-item>
                        <f!compare><f!alter-list>
                        INCREMENT THE <f!incre-spec> <f!incre-prep> <f!nine-digit> POINTS

<f!intro>::=            I WANT TO
                        FOR EACH <f!iter-item>

<f!iter-item>::=        PHRASE
                        PHONE
                        PHONEME
                        SEGMENT
                        WINDOW
                        FUNCTION
                        TIME
                        POSITION
                        SENTENCE
                        UTTERANCE

<f!param-phr>::=        <f!param-spec>
                        <f!param-phr><f!prep><f!param-spec>

<f!param-spec>::=       FILE NUMBER <f!nine-digit>
                        UTTERANCE NUMBER <f!nine-digit>
                        A <f!wind-type> WINDOW OF <f!nine-digit> POINTS
                        A <f!freq-spec> OF <f!nine-digit> HERTZ
```

```
                        A <f!res-type> RESOLUTION OF <f!nine-digit><f!res-unit>
                        <f!nine-digit> COEFFICIENTS
                        AN ORDER OF <f!nine-digit>
                        START TIME <f!num>
                        END TIME <f!nine-digit>
                        A <f!emph-type> OF <f!nine-digit><f!db> PER OCTAVE
                        A SCALE FACTOR OF <f!nine-digit>
                        A FLOOR OF <f!num>
                        A CEILING OF <f!nine-digit>

<f!prep>::=             OF
                        TO
                        WITH
                        ON

<f!wind-type>::=        HAMMING
                        HANNING
                        BLACKWELL
                        RECTANGULAR
                        TRIANGULAR

<f!freq-spec>::=        FREQUENCY
                        <f!freq-type> FREQUENCY
                        BANDWIDTH

<f!freq-type>::=        CENTER
                        CUTOFF
                        LOW PASS
                        HIGH PASS

<f!meth-type>::=        <f!name> 'S
                        THE <f!meth-kind>

<f!name>::=             ITAKURA
                        MARKEL
                        PRONY
                        ATAL
                        ROBINSON
                        SCHAFFER AND RABINER
                        FANT
                        NEWTON
                        BAIRSTOW

<f!meth-kind>::=        AUTOCORRELATION
                        COVARIANCE
                        PEAK PICKING
                        ROOT FINDING

<f!res-type>::=         TIME
                        FREQUENCY

<f!res-unit>::=         HERTZ
                        CYCLES PER SECOND
                        MICROSECONDS
                        MILLISECONDS
                        CENTISECONDS
                        POINTS

<f!emph-type>::=        PRE-EMPHASIS
                        POST-EMPHASIS
```

```
12400
12500  <f!db>::=            DECIBELS
12600                       DB
12700
12800  <f!compute>::=       COMPUTE
12900                       CALCULATE
13000                       FIND
13100                       GET
13200                       TAKE
13300                       CONSIDER
13400
13800  <f!func-phr>::=      THE <f!comp-func>
13900                       THE AUTOCORRELATION FUNCTION
14000                       THE COVARIANCE FUNCTION
14100                       THE FFT
14200                       THE FAST FOURIER TRANSFORM
14300                       THE FOURIER TRANSFORM
14400                       THE HILBERT TRANSFORM
14600                       THE LINEAR PREDICTION COEFFICIENTS
14700                       THE LINEAR PREDICTION FILTER
14800                       THE INVERSE FILTER
14900                       THE SPECTRUM
15000                       THE CEPSTRUM
15100                       THE <f!spec-adj> SPECTRUM
15150                       THE ROOTS
15200
15300  <f!comp-func>::=     <f!func-part>
15400                       <f!func-part> OF <f!func-phr>
15500
15600  <f!func-part>::=     ROOTS
15700                       PEAKS
15800                       IMAGINARY PART
15900                       REAL PART
16000                       LOGARITHM
16100                       ABSOLUTE VALUE
16200
16300  <f!plot>::=          PLOT
16400                       DISPLAY
16500                       SHOW
16600
16700  <f!plot-item>::=     THE SPECTROGRAM
16710                       THE SPECTROGRAM <f!prep><f!param-phr>
16800                       THE WAVEFORM
16900                       THE FORMANT TRACKS
17000                       THE FUNCTION
17100                       <f!func-phr>
17200
17300  <f!spec-adj>::=      SMOOTHED
17400                       <f!smth-meth> SMOOTHED
17500                       <f!spec-meth>
17600
17700  <f!spec-meth>::=     CEPSTRAL
17800                       LINEAR PREDICTIVE
17900                       INVERSE FILTERED
18000                       FFT
18100                       FAST FOURIER TRANSFORM
18200                       FOURIER
18300
18400  <f!smth-meth>::=     CEPSTRALLY
18500                       LINEAR PREDICTION
```

```
                        INVERSE FILTER
                        LPC

<f!compare>::=          COMPARE
                        LOOK AT
                        CONSIDER

<f!alter-list>::=       <f!meth-type> METHOD <f!meth-conj><f!meth-type> METHOD
                        ANOTHER METHOD OF <f!form-task>
                        <f!form-task> METHODS
                        <f!form-task> WITH DIFFERENT PARAMETERS

<f!meth-conj>::=        AND
                        WITH

<f!form-task>::=        FORMANT ESTIMATION
                        SPECTRAL SMOOTHING
                        IMAGE ENHANCEMENT
                        ROOT FINDING
                        LINEAR PREDICTION

<f!incre-prep>::=       BY
                        IN STEPS OF

<f!incre-spec>::=       WINDOW
                        STARTING TIME

; This is the number sub-grammar.
; It is used by most of the task sub-grammars.

<f!num>::=              <f!nine-digit>
                        ZERO

<f!nine-digit>::=       <f!six-digit>
                        <f!three-digit> MILLION <f!six-digit>

<f!six-digit>::=        <f!three-digit>
                        <f!three-digit> THOUSAND <f!three-digit>

<f!three-digit>::=      <f!two-digit>
                        <f!digit> HUNDRED <f!two-digit>
                        <f!digit> HUNDRED

<f!two-digit>::=        <f!digit>
                        <f!teen>
                        <f!tens><f!digit>
                        <f!tens>

<f!tens>::=             TWENTY
                        THIRTY
                        FOURTY
                        FIFTY
                        SIXTY
                        SEVENTY
                        EIGHTY
                        NINETY

<f!teen>::=             TEN
                        ELEVEN
                        TWELVE
```

```
                     THIRTEEN
                     FOURTEEN
                     FIFTEEN
                     SIXTEEN
                     SEVENTEEN
                     EIGHTEEN
                     NINETEEN

<f!digit>::=         ONE
                     TWO
                     THREE
                     FOUR
                     FIVE
                     SIX
                     SEVEN
                     EIGHT
                     NINE
```

```
;  SYNTAX FOR AP VOICE NEWS QUERY SYSTEM.  28 TERMINAL SYMBOLS (WORDS).

<QUERY>::=               [ <REQUEST> ]

<REQUEST>::=             LET <PRONOUNA> HAVE <COLL-SUM>
                         GIVE <PRONOUNB><NOUN-PHRASE>
                         GIVE <PRONOUNB><COLL-SUM>
                         TELL <PRONOUNC><COLL-SUM>
                         TELL <PRONOUNC><QUANTIFIER><NOUN-PHRASE>
                         TELL <PRONOUNC><TELL-QUAN><SUM-PHRASE>

<COLL-SUM>::=            <SUM-PHRASE>
                         ALL <SUM-PHRASE>
                         SEX

<SUM-PHRASE>::=          THE <SUMMARIESB>
                         THE <SUMMARIESA> AND <SUMMARIESB>

<SUMMARIESA>::=          STORIES
                         HEADLINES
                         SUMMARY

<SUMMARIESB>::=          STORIES
                         HEADLINES
                         SUMMARY

<TELL-QUAN>::=           <QUANTIFIER>
                         ABOUT ALL
                         ALL

<PRONOUNA>::=            ME
                         US

<PRONOUNB>::=            ME
                         US

<PRONOUNC>::=            ME
                         US

<QUANTIFIER>::=          ALL ABOUT
                         ABOUT

<NOUN-PHRASE>::=         <NOUNA> AND <NOUNB>
                         <NOUNA> OR <NOUNB>
                         <NOUNB>

<NOUNA>::=               FRANCE
                         AIRPLANE HIJACKING
                         HIJACKING
                         CHINA
                         ISRAEL
                         MURDER
                         NIXON
                         RAPE
                         RUSSIA
                         SEX
                         AIRPLANES
                         VIETNAM
                         WAR
                         THE VIETNAM WAR
```

```
                    WATERGATE
                    THE WATERGATE

<NOUNB>::=          FRANCE
                    AIRPLANE HIJACKING
                    HIJACKING
                    CHINA
                    ISRAEL
                    MURDER
                    NIXON
                    RAPE
                    RUSSIA
                    SEX
                    AIRPLANES
                    VIETNAM
                    WAR
                    THE VIETNAM WAR
                    WATERGATE
                    THE WATERGATE
```

```
   GRAMMAR FOR CHESS

<move>::=        [ <moveb> ]

<moveb>::=       <movea><check-word>
                 <movea>

<movea>::=       <pce-loc><motion><loca>
                 <pce-loc><takes><pce-loca>
                 <castle-move>

<pce-loc>::=     <piece>
                 <piece> ON <loc>

<loc>::=         <pieceb><square>

<pce-loca>::=    <piecec>
                 <piecec> ON <loca>

<loca>::=        <pieced><squarea>

<piece>::=       <royal>
                 <royal><man>
                 <man>

<man>::=         <bnr>
                 <bnr> PAWN
                 PAWN

<pieceb>::=      <royalb>
                 <royalb><manb>
                 <manb>

<manb>::=        <bnrb>
                 <bnrb> PAWN
                 PAWN

<piecec>::=      <royalc>
                 <royalc><manc>
                 <manc>

<manc>::=        <bnrc>
                 <bnrc> PAWN
                 PAWN

<pieced>::=      <royald>
                 <royald><mand>
                 <mand>

<mand>::=        <bnrd>
                 <bnrd> PAWN
                 PAWN

<royal>::=       KING
                 QUEEN

<bnr>::=         BISHOP
                 KNIGHT
                 ROOK
```

```
<royald>::=     KING
                QUEEN

<bnrd>::=       BISHOP
                KNIGHT
                ROOK

<royalb>::=     KING
                QUEEN

<bnrb>::=       BISHOP
                KNIGHT
                ROOK

<royalc>::=     KING
                QUEEN

<bnrc>::=       BISHOP
                KNIGHT
                ROOK

<square>::=     ONE
                TWO
                THREE
                FOUR
                FIVE
                SIX
                SEVEN
                EIGHT

<squarea>::=    ONE
                TWO
                THREE
                FOUR
                FIVE
                SIX
                SEVEN
                EIGHT

<motion>::=     TO
                MOVES-TO
                GOES-TO

<takes>::=      TAKES
                CAPTURES

<castle-move>::=        CASTLE
                CASTLE ON <royale> SIDE
                CASTLE <royale> SIDE

<royale>::=     KING
                QUEEN

<check-word>::= CHECK
                MATE
```

```
            BNF FOR THE DOCTOR INTERVIEW.  76 TERMINAL WORDS.

 <HEAD>::= [ <SENTENCE> ]

 <SENTENCE>::= <INTEROGB> <HABIT-VERB>
        <INTEROGC> <SYMPTOM>
        <INTEROGD> <SYMPTOM> <ADJ>
        <INTEROGE> <SYMPTOMS> <ADJ>
        <INTEROGG> <PHYS-COND>
        <INTEROGG> <PERSONAL-STATE>
        <INTEROGH> <VERBA> <AILMENT>
        <INTEROGH> <VERBB> <PARTICIPIAL>
        <W> <INTEROGF> <PARTICIPIAL>
        <INTEROGD> <PERSONAL-NOUN> <PERSONAL-ADJ>

 <W>::= WHERE
        WHEN

 <QUANTIFIER>::= OFTEN
        LONG
        FREQUENTLY
        MUCH

 <INTEROGA>::= HOW
        HOW <QUANTIFIER>

 <INTEROGB>::= DO YOU
        <INTEROGA> DO YOU

 <INTEROGC>::= WHERE IS THE

 <INTEROGD>::= IS THE
        IS YOUR

 <INTEROGE>::= ARE THE
        ARE YOUR

 <INTEROGF>::= WERE YOU
        WERE YOU EVER

 <INTEROGG>::= ARE YOU
        <INTEROGF>

 <INTEROGH>::= HAVE YOU
        <INTEROGA> HAVE YOU

 <VERBA>::= HAD
        EVER HAD

 <VERBB>::= BEEN
        EVER BEEN

 <HABIT-VERB>::= SMOKE
        DRINK
        OVEREAT
        SMOKE <SMOKEY-ADJ>

 <SMOKEY-ADJ>::= CIGARETTES
        POT
        GRASS
```

```

<SYMPTOM>::= PAIN
        NUMBNESS
        NAUSEA
        DIZZINESS
        BLEEDING

<SYMPTOMS>::= HEADACHES
        PAINS
        CRAMPS
        CHEST PAINS
        LESIONS

<AILMENT>::= MUMPS
        MEASLES
        CHICKEN-POX
        TUBERCULOSIS
        ASTHMA
        GONORRHEA
        CLOUDY URINE
        SURGERY
        AN OPERATION

<ADJ>::= SEVERE
        MILD
        BAD
        CONTINUOUS
        SHARP
        SERIOUS

<PHYS-COND>::= SICK
        ILL
        IN PAIN
        FEVERISH
        DEAD

<PERSONAL-STATE>::= AFRAID OF SURGERY
        CASTRATED

<PERSONAL-NOUN>::= URINE
        HEAD


<PERSONAL-ADJ>::= CLOUDY
        ATTACHED

<PARTICIPIAL>::= HOSPITALIZED
        CIRCUMCISED
        ANESTHETIZED
        CASTRATED
        AFRAID OF SURGERY
        IMMUNIZED
        INJURED
        SERIOUS

```

```
<sentence>::=     [ <request> ]

<request>::=      COMPUTE <func-phr>
                  USE <param-phr>

<func-phr>::=     <function>
                  <function> USING <param-phr>

<function>::=     THE <name> TRANSFORM

<name>::=         HILBERT
                  FOURIER

<param-phr>::=    <param-spec>
                  <param-spec> WITH <param-phr>

<param-spec>::= A LENGTH OF FIVE HUNDRED TWELVE POINTS
                  A HAMMING WINDOW
```

```
181
182
44
<sentence>::=  1       -1      0
[       2      181      1
               1        1000
<request>      3        -2      1
               2        1000
]       4      182      1
               11       1000
ENDOF<sentence> 5       -1      1
               4        1000
<request>::=   6        -2      1
               2        1000
COMPUTE 7      291      1
               6        1000
<func-phr>     8        -3      1
               7        1000
USE     9      222      1
               6        1000
<param-phr>    10       -6      1
               9        1000
ENDOF<request> 11       -2      2
               17       500
               32       500
<func-phr>::=  12       -3      1
               7        1000
<function>     13       -4      1
               12       1000
<function>     14       -4      1
               12       1000
USING   15     252      1
               22       1000
<param-phr>    16       -6      1
               15       1000
ENDOF<func-phr> 17      -3      2
               22       500
               32       500
<function>::=  18       -4      2
               12       500
               12       500
THE     19     156      1
               18       1000
<name>  20     -5       1
               19       1000
TRANSFORM      21       300     1
               26       1000
ENDOF<function> 22      -4      1
               21       1000
<name>::=      23       -5      1
               19       1000
HILBERT 24     301      1
               23       1000
FOURIER 25     299      1
               23       1000
ENDOF<name>    26       -5      2
               24       500
               25       500
<param-phr>::= 27       -6      3
               9        333
```

```
                  15      333
                  30      334
<param-spec>      28      -7      1
                  27      1000
<param-spec>      29      -7      1
                  27      1000
WITH      30      251     1
                  44      1000
<param-phr>       31      -6      1
                  30      1000
ENDOF<param-phr>          32      -6      2
                  44      500
                  32      500
<param-spec>::=   33      -7      2
                  27      500
                  27      500
A         34      1       1
                  33      1000
LENGTH    35      565     1
                  34      1000
OF        36      117     1
                  35      1000
FIVE      37      58      1
                  36      1000
HUNDRED   38      338     1
                  37      1000
TWELVE    39      349     1
                  38      1000
POINTS    40      225     1
                  39      1000
A         41      1       1
                  33      1000
HAMMING   42      253     1
                  41      1000
WINDOW    43      232     1
                  42      1000
ENDOF<param-spec>         44      -7      2
                  40      500
                  43      500
```

```
  2
  4
135
  1 -       0      0 "NULL"      0    900     0
  2 -       0    101 [       1      0    900
        1    100
  3 -       0      0 "NULL"      1    900     0
        2   1000
  4 -       0    102 ]       1      0    900
       23    100
  5 -       0      0 "NULL"      1    900     0
        4   1000
  6 -       0      0 "NULL"      1    900     0
        2   1000
  7 -       0    291 COMPUTE      1      0    900
        6    100
  8 K       5    291 COMPUTE      1      0    900
        7    100
  9 AH     24    291 COMPUTE      1      0    900
        8    100
 10 M      13    291 COMPUTE      1      0    900
        9    100
 11 -       0    291 COMPUTE      1      0    900
       10    100
 12 P       1    291 COMPUTF      1      0    900
       11    100
 13 Y      18    291 COMPUTE      1      0    900
       12    100
 14 UW     19    291 COMPUTE      1      0    900
       13    100
 15 -       0    291 COMPUTE      1      0    900
       14    100
 16 T       3    291 COMPUTE      1      0    900
       15    100
 17 -       0      0 "NULL"      1    900     0
       16   1000
 18 -       0    222 USE      1      0    900
        6    100
 19 Y      18    222 USE      1      0    900
       18    100
 20 UW     19    222 USE      1      0    900
       19    100
 21 S      10    222 USE      1      0    900
       20    100
 22 -       0      0 "NULL"      1    900     0
       21   1000
 23 -       0      0 "NULL"      2    900     0
       34    500
       78    500
 24 -       0      0 "NULL"      1    900     0
       16   1000
 25 -       0      0 "NULL"      1    900     0
       24   1000
 26 -       0      0 "NULL"      1    900     0
       24   1000
 27 -       0    252 USING      1      0    900
       51    100
 28 Y      18    252 USING      1      0    900
       27    100
 29 UW     19    252 USING      1      0    900
```

```
        28    100
30 S      10     252 USING       1      0    900
        29    100
31 IH     28     252 USING       1      0    900
        30    100
32 NX     15     252 USING       1      0    900
        31    100
33 -       0       0 "NULL"      1    900      0
        32   1000
34 -       0       0 "NULL"      2    900      0
        51    500
        78    500
35 -       0       0 "NULL"      2    900      0
        24    500
        24    500
36 -       0     156 THE       1      0    900
        35    100
37 OH      9     156 THE       1      0    900
        36    100
38 AX     30     156 THE       1      0    900
        37    100
39 -       0       0 "NULL"      1    900      0
        38   1000
40 -       0     300 TRANSFORM      1      0    900
        69    100
41 T       3     300 TRANSFORM      1      0    900
        40    100
42 ER     25     300 TRANSFORM      1      0    900
        41    100
43 AE     26     300 TRANSFORM      1      0    900
        42    100
44 N      14     300 TRANSFORM      1      0    900
        43    100
45 -       0     300 TRANSFORM      1      0    900
        44    100
46 S      10     300 TRANSFORM      1      0    900
        45    100
47 F       7     300 TRANSFORM      1      0    900
        46    100
48 AO     22     300 TRANSFORM      1      0    900
        47    100
49 ER     25     300 TRANSFORM      1      0    900
        48    100
50 M      13     300 TRANSFORM      1      0    900
        49    100
51 -       0       0 "NULL"      1    900      0
        50   1000
52 -       0       0 "NULL"      1    900      0
        38   1000
53 -       0     301 HILBERT      1      0    900
        52    100
54 HH     12     301 HILBERT      1      0    900
        53    100
55 IH     28     301 HILBERT      1      0    900
        54    100
56 L      17     301 HILBERT      1      0    900
        55    100
57 -       0     301 HILBERT      1      0    900
        56    100
58 B       2     301 HILBERT      1      0    900
```

```
          57    100
59 ER       25     301 HILBERT       1      0     900
          58    100
60 -         0     301 HILBERT       1      0     900
          59    100
61 T         3     301 HILBERT       1      0     900
          60    100
62 -         0     299 FOURIER       1      0     900
          52    100
63 F         7     299 FOURIER       1      0     900
          62    100
64 AO       22     299 FOURIER       1      0     900
          63    100
65 ER       25     299 FOURIER       1      0     900
          64    100
66 IY       29     299 FOURIER       1      0     900
          65    100
67 EH       27     299 FOURIER       1      0     900
          66    100
68 IH       28     299 FOURIER       1      0     900
          67    100
69 -         0       0 "NULL"       2     900      0
          61    500
          68    500
70 -         0       0 "NULL"       3     900      0
          21    333
          32    333
          76    334
71 -         0       0 "NULL"       1     900      0
          70   1000
72 -         0       0 "NULL"       1     900      0
          70   1000
73 -         0     251 WITH       1      0     900
         135    100
74 W        16     251 WITH       1      0     900
          73    100
75 IH       28     251 WITH       1      0     900
          74    100
76 F         7     251 WITH       1      0     900
          75    100
77 -         0       0 "NULL"       1     900      0
          76   1000
78 -         0       0 "NULL"       2     900      0
         135    500
          78    500
79 -         0       0 "NULL"       2     900      0
          70    500
          70    500
80 -         0       1 A       1      0     900
          79    100
81 AX       30       1 A       1      0     900
          80    100
82 -         0     565 LENGTH       1      0     900
          81    100
83 L        17     565 LENGTH       1      0     900
          82    100
84 AX       30     565 LENGTH       1      0     900
          83    100
85 NX       15     565 LENGTH       1      0     900
          84    100
```

```
 86 -       0   565 LENGTH     1     0    900
       85   100
 87 F       7   565 LENGTH     1     0    900
       86   100
 88 -       0   117 OF         1     0    900
       87   100
 89 AO     22   117 OF         1     0    900
       88   100
 90 V       8   117 OF         1     0    900
       89   100
 91 -       0    58 FIVE       1     0    900
       90   100
 92 F       7    58 FIVE       1     0    900
       91   100
 93 AA     23    58 FIVE       1     0    900
       92   100
 94 AX     30    58 FIVE       1     0    900
       93   100
 95 V       8    58 FIVE       1     0    900
       94   100
 96 -       0   338 HUNDRED    1     0    900
       95   100
 97 HH     12   338 HUNDRED    1     0    900
       96   100
 98 AH     24   338 HUNDRED    1     0    900
       97   100
 99 N      14   338 HUNDRED    1     0    900
       98   100
100 -       0   338 HUNDRED    1     0    000
       99   100
101 D       4   338 HUNDRED    1     0    900
      100   100
102 ER     25   338 HUNDRED    1     0    900
      101   100
103 EH     27   338 HUNDRED    1     0    900
      102   100
104 -       0   338 HUNDRED    1     0    900
      103   100
105 D       4   338 HUNDRED    1     0    900
      104   100
106 -       0   349 TWELVE     1     0    900
      105   100
107 T       3   349 TWELVE     1     0    900
      106   100
108 W      16   349 TWELVE     1     0    900
      107   100
109 EH     27   349 TWELVE     1     0    000
      108   100
110 L      17   349 TWELVE     1     0    900
      109   100
111 V       8   349 TWELVE     1     0    900
      110   100
112 -       0   225 POINTS     1     0    900
      111   100
113 P       1   225 POINTS     1     0    900
      112   100
114 AO     22   225 POINTS     1     0    900
      113   100
115 IH     2[illegible]   225 POINTS     1     0    900
      114   100
```

```
116 N       14    225 POINTS       1      0    900
        115    100
117 -        0    225 POINTS       1      0    900
        116    100
118 T        3    225 POINTS       1      0    900
        117    100
119 S       10    225 POINTS       1      0    900
        118    100
120 -        0      1 A      1      0    900
         79    100
121 AX      30      1 A      1      0    900
        120    100
122 -        0    253 HAMMING      1      0    900
        121    100
123 HH      12    253 HAMMING      1      0    900
        122    100
124 AE      26    253 HAMMING      1      0    900
        123    100
125 M       13    253 HAMMING      1      0    900
        124    100
126 IH      28    253 HAMMING      1      0    900
        125    100
127 NX      15    253 HAMMING      1      0    900
        126    100
128 -        0    232 WINDOW      1      0    900
        127    100
129 W       16    232 WINDOW      1      0    900
        128    100
130 IH      28    232 WINDOW      1      0    900
        129    100
131 N       14    232 WINDOW      1      0    900
        130    100
132 -        0    232 WINDOW      1      0    900
        131    100
133 D        4    232 WINDOW      1      0    900
        132    100
134 OW      21    232 WINDOW      1      0    900
        133    100
135 -        0      0 "NULL"      2    900      0
        119    500
        134    500
```

2: JKB2: USE A HAMMING WINDOW OF FIVE HUNDRED TWELVE POINTS

95:	0	0	0	0	0	0	0	0	0	0	0	0
96:	0	0	0	0	0	0	0	0	0	0	0	0
97:	0	0	0	0	0	0	0	0	0	0	0	0
98:	0	0	0	0	0	0	0	0	0	0	1	0
99:	0	0	0	0	0	0	0	0	0	0	0	0
100:	0	0	0	0	0	0	0	0	0	0	0	0
101:	0	0	0	0	0	0	0	0	0	0	0	0
102:	0	0	0	0	0	0	0	0	0	0	6	0
103:	0	0	0	0	0	0	0	0	0	0	1	0
104:	0	0	0	0	0	0	0	0	0	0	0	0
105:	0	0	0	0	0	0	0	0	0	0	0	0
106:	0	0	0	0	0	0	0	0	0	0	0	0
107:	0	0	0	0	0	0	0	0	0	0	0	0
108:	0	0	0	0	0	0	0	50	0	0	5	4
109:	0	16	0	5	0	0	219	21	304	90	52	12
110:	8	34	0	4	0	0	257	34	253	85	63	12
111:	27	20	0	7	0	1	205	58	269	62	143	46
112:	28	25	0	9	0	4	172	62	282	78	170	52
113:	32	33	12	14	0	5	152	64	230	85	191	84
114:	25	46	33	21	7	10	158	72	265	76	164	99
115:	18	50	33	37	16	14	158	108	251	76	117	115
116:	16	61	31	46	22	22	144	100	241	66	159	119
117:	15	60	31	49	39	24	149	109	246	57	135	123
118:	20	64	33	55	50	30	130	87	250	46	151	114
119:	21	65	34	55	97	34	150	68	246	48	89	108
120:	26	73	41	58	114	44	145	48	226	30	93	103
121:	25	98	48	66	125	54	159	41	175	20	68	95
122:	32	101	48	65	143	57	161	34	196	20	30	91
123:	32	116	42	70	141	56	167	32	146	21	43	99
124:	32	122	54	74	154	58	145	23	141	25	30	107
125:	38	132	36	86	157	53	96	19	191	25	30	105
126:	36	160	40	117	157	52	64	25	149	26	35	92
127:	43	169	47	135	166	58	52	24	116	23	35	86
128:	42	164	46	166	160	60	69	25	91	19	35	81
129:	44	165	46	180	151	66	71	20	74	19	35	80
130:	34	154	53	201	138	63	80	19	77	18	35	69
131:	31	127	62	200	159	65	95	18	40	19	43	67
132:	26	118	66	172	184	66	92	20	59	20	35	65
133:	30	97	57	140	193	58	84	19	118	21	47	62
134:	25	90	65	123	166	54	119	30	147	22	39	51
135:	30	101	78	121	232	54	107	28	68	24	35	41
136:	42	104	90	104	287	56	58	22	38	24	43	32
137:	37	90	90	60	233	42	0	10	192	37	52	30
138:	45	82	15	33	27	21	0	3	337	79	94	23
139:	29	37	1	5	0	0	0	0	371	58	243	11
140:	31	25	0	4	0	0	0	0	255	46	292	10
141:	0	10	0	0	0	0	0	0	377	30	318	10
142:	0	1	0	0	0	0	0	0	262	39	358	10
143:	0	0	1	0	0	0	0	0	389	25	403	12
144:	0	0	1	0	0	0	0	0	387	33	283	10
145:	0	0	0	0	0	0	0	0	0	0	5	5
146:	263	87	0	105	0	70	0	17	0	0	22	4
147:	0	93	0	93	0	62	0	15	0	0	43	4
148:	0	100	0	300	0	50	0	0	0	0	9	2
149:	0	0	0	50	0	0	0	0	0	0	1	1
150:	0	0	0	0	0	0	0	0	0	0	1	0
151:	0	0	0	0	0	0	0	0	0	0	1	0
152:	0	0	0	0	0	0	0	0	0	0	1	0
153:	0	0	0	0	0	0	0	0	0	0	1	0

154:	0	0	0	0	0	0	0	0	0	0	1	0
155:	0	0	0	0	0	0	0	0	0	0	1	0
156:	0	0	0	25	0	25	0	89	0	25	60	5
157:	0	3	0	30	43	43	123	96	143	20	97	52
158:	0	7	91	183	111	47	143	75	67	25	60	63
159:	41	27	93	174	63	54	118	59	77	39	56	83
160:	36	27	75	177	97	41	134	52	94	41	48	90
161:	33	49	85	220	47	42	108	69	44	49	51	89
162:	52	60	67	190	62	31	89	62	123	54	43	86
163:	51	68	64	151	81	27	122	51	145	54	35	83
164:	60	89	87	138	47	23	111	54	107	69	43	72
165:	46	92	73	104	29	22	133	49	162	63	55	75
166:	38	78	59	77	42	20	168	60	193	49	73	75
167:	40	66	52	54	25	18	247	88	94	67	104	94
168:	22	58	52	46	32	15	235	89	91	71	149	91
169:	39	51	58	46	0	9	197	92	152	72	122	84
170:	83	55	62	104	0	24	181	52	87	34	72	17
171:	28	48	4	53	0	24	207	57	82	43	76	17
172:	0	14	0	37	0	23	242	42	32	37	105	17
173:	0	5	5	30	0	30	131	70	35	40	115	14
174:	0	0	3	18	0	18	255	62	62	29	137	14
175:	0	0	0	14	0	17	338	63	0	21	138	17
176:	0	4	0	17	0	22	150	53	0	26	151	13
177:	0	11	0	27	11	31	169	35	83	39	135	14
178:	28	28	63	84	60	60	124	86	124	40	65	37
179:	27	12	59	113	84	59	61	78	176	49	65	172
180:	16	13	44	100	68	59	100	88	209	48	114	169
181:	18	17	52	115	71	69	105	93	173	58	76	158
182:	22	17	45	109	75	67	130	65	206	57	85	126
183:	25	19	54	122	79	69	117	51	175	67	81	121
184:	22	17	50	117	80	62	122	32	215	60	89	137
185:	27	17	62	135	76	83	105	38	175	60	77	146
186:	21	16	54	127	70	104	118	38	179	43	97	154
187:	26	18	50	122	66	113	111	51	183	43	85	151
188:	24	21	56	107	70	111	137	52	192	32	77	145
189:	31	29	63	120	107	128	164	60	77	11	64	118
190:	46	37	59	155	186	160	150	42	5	6	56	32
191:	28	63	14	109	215	140	175	51	0	8	35	32
192:	38	71	35	38	178	73	234	43	17	20	60	38
193:	29	67	69	38	137	64	264	60	48	15	67	38
194:	25	70	37	34	138	56	265	53	74	17	80	50
195:	14	52	40	104	88	38	156	53	242	33	92	88
196:	14	59	52	104	59	28	145	46	266	45	77	106
197:	14	51	54	99	56	20	167	36	256	44	100	96
198:	16	53	61	98	58	20	161	40	253	48	80	89
199:	17	56	64	92	71	19	149	39	261	49	72	80
200:	22	70	51	98	57	22	215	39	190	33	81	52
201:	48	114	85	126	55	21	277	34	19	24	43	36
202:	101	238	190	178	0	35	0	17	0	0	18	22
203:	115	238	207	115	0	23	0	7	0	0	18	20
204:	135	279	126	126	0	18	0	0	0	0	18	17
205:	234	375	0	93	0	15	0	0	0	0	13	5
206:	283	264	0	94	0	37	0	0	0	0	13	4
207:	0	147	0	205	0	58	0	0	0	0	13	7
208:	0	135	27	189	0	81	0	0	0	0	9	12
209:	263	115	105	157	0	73	0	0	0	0	13	14
210:	120	76	125	96	149	76	0	4	0	0	35	38
211:	83	80	132.	98	213	106	0	2	0	0	39	58
212:	51	94	83	117	161	158	31	8	0	10	63	96
213:	25	61	39	96	111	164	82	66	76	24	92	149

2: JKB2: USE A HAMMING WINDOW OF FIVE HUNDRED TWELVE POINTS

95:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
96:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
97:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
98:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	1	3925
99:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
100:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
101:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
102:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
103:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	1	3925
104:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
105:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
106:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
107:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	0	4018
108:	- 1	F 29	K 162	HH 49	V 36	S 41	F 28	2541	4173
109:	Y 84	G 27	D 19	IY 143	D 17	P 12	P 8	15497	19760
110:	Y 84	P 8	D 17	G 27	P 12	IY 143	IY 145	7952	16759
111:	Y 84	D 19	D 17	SH 42	N 65	T 15	IY 143	5772	11438
112:	D 19	Y 84	UW 94	IY 143	SH 42	N 65	T 15	9944	12102
113:	UW 94	N 65	IY 143	D 17	Y 84	IH 141	T 15	7324	8440
114:	IY 143	UW 94	N 65	Y 84	IH 141	D 19	D 17	5798	6052
115:	IY 143	UW 94	N 65	IH 141	UW 86	IH 137	Y 84	4681	8643
116:	UW 94	IY 143	IH 141	N 65	IH 137	IY 142	UW 86	3845	7153
117:	UW 94	IY 143	UW 86	IH 141	IH 137	N 65	IY 142	5069	6603
118:	UW 94	UW 86	IY 143	N 65	IH 137	IH 141	ER 123	3932	8000
119:	UW 86	ER 123	IY 143	UW 94	IH 137	AX 150	N 65	2253	8575
120:	UH 86	ER 123	AX 151	UW 94	IH 137	AX 150	IY 143	3009	5253
121:	AX 151	UW 86	AX 149	AX 147	ER 123	UW 88	UW 91	5418	8832
122:	AX 151	AX 147	UW 88	UW 86	AX 149	ER 123	UW 91	4688	9942
123:	AX 151	UW 91	UW 88	AX 149	AX 147	Y 165	ER 122	5697	7339
124:	UW 91	AX 151	UW 88	AX 149	AX 147	UW 93	ER 122	7379	8207
125:	UW 88	AX 151	UW 93	ER 122	AX 149	L 80	UW 86	13226	15364
126:	UW 88	UW 93	UW 91	AX 149	ER 122	AX 151	L 80	12905	14210
127:	UW 88	UW 93	L 83	L 82	UW 91	V 33	L 81	15452	17811
128:	UW 88	L 82	UW 93	L 83	V 33	AX 154	UW 91	13468	13786
129:	L 82	UW 88	V 33	L 83	UW 93	AX 154	AO 107	9821	15039
130:	L 82	UW 88	AO 107	AX 154	UW 93	V 33	L 83	6763	13411
131:	L 82	AX 154	AO 107	ER 120	V 33	UW 88	L 83	6554	11203
132:	L 82	ER 120	AX 154	UW 88	V 33	UW 91	NX 70	11697	12394
133:	UW 88	UW 91	AX 151	AX 155	AX 149	UW 93	L 82	9854	17034
134:	UW 88	AX 151	AX 149	UW 91	AX 147	UW 93	Y 165	4751	7173
135:	AX 152	ER 120	UW 91	M 55	NX 70	M 53	UW 88	12474	14788
136:	M 55	ER 125	HH 45	M 53	HH 47	AX 152	- 4	13305	14771
137:	L 80	AX 155	AX 151	UW 88	ER 125	HH 45	HH 47	27523	36606
138:	F 30	Y 163	D 20	T 14	L 80	IY 143	D 19	23654	26352
139:	T 14	S 38	S 40	S 39	F 30	D 19	D 20	4633	17775
140:	S 40	T 14	S 38	F 30	D 20	T 13	D 19	2359	20085
141:	S 38	T 14	S 39	S 40	D 19	F 30	D 20	3061	10319
142:	S 40	S 38	T 14	S 39	F 30	D 20	D 19	6336	18190
143:	S 38	S 39	T 14	S 40	D 19	SH 43	T 15	2094	2125
144:	T 14	S 38	S 39	S 40	D 19	F 30	D 20	5596	7138
145:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	50	3578
146:	N 62	- 3	N 59	W 75	N 66	M 52	N 58	10583	20927
147:	DH 37	K 162	HH 50	V 36	HH 49	- 6	D 16	6219	8257
148:	W 78	W 73	AO 107	L 82	W 77	AO 109	L 79	7605	35088
149:	- 1	F 29	K 162	V 36	HH 49	S 41	F 28	2502	6422
150:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	1	3925
151:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	1	3925
152:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	1	3925
153:	- 1	F 29	V 36	S 41	K 162	F 28	HH 49	1	3925

154: - 1	F 29	V 36	S 41	K 162	F 28	HH 49	1	3925
155: - 1	F 29	V 36	S 41	K 162	F 28	HH 49	1	3925
156: HH 49	K 162	F 29	F 28	S 41	- 1	V 36	3294	4596
157: D 17	D 18	G 27	N 65	ER 123	IH 137	T 13	10583	16399
158: AX 154	AX 149	ER 161	EH 160	HH 48	UW 88	AE 167	19735	21255
159: EH 160	AX 149	ER 161	UW 88	AE 167	L 82	AX 154	16568	16827
160: AX 149	UW 88	ER 161	AX 154	EH 160	AE 167	UW 93	11725	13214
161: EH 160	L 82	AO 107	AX 154	AE 167	L 83	ER 161	13564	17812
162: UW 88	UW 93	AX 149	AX 146	L 82	AE 167	OW 99	15823	16402
163: UW 88	AX 149	UW 93	AX 146	IH 138	AX 151	DW 104	8406	9933
164: NX 71	UW 88	AX 149	UW 93	L 83	IH 138	L 82	13955	14970
165: AX 150	AX 149	UW 88	N 65	IH 138	IY 144	UW 86	16371	16622
166: N 65	AX 150	UW 86	IY 143	IH 137	IY 144	UW 94	8937	9525
167: IY 145	N 65	Y 164	D 17	IH 137	N 60	P 9	17482	20199
168: N 65	D 17	P 9	IH 137	Y 164	UW 94	K 23	16857	21880
169: N 65	D 17	IH 137	UW 94	Y 84	IH 141	IY 143	4580	12643
170: M 56	NX 71	IY 145	Y 85	N 65	D 17	IY 144	17914	18212
171: D 17	IY 145	Y 164	HH 44	K 23	D 18	P 9	13998	14116
172: HH 44	Y 164	K 23	G 26	K 24	N 60	D 18	3777	5781
173: K 23	HH 44	D 18	T 13	F 28	HH 49	D 17	6377	11433
174: Y 164	HH 44	K 24	K 23	P 9	D 18	D 17	5728	6684
175: Y 164	G 26	N 60	HH 44	K 24	K 23	P 9	5068	8557
176: K 23	HH 44	D 18	T 13	Y 164	F 28	P 9	3642	5194
177: D 18	HH 44	K 23	D 17	P 9	T 13	K 24	3706	9799
178: P 10	AX 149	UW 88	ER 123	N 65	AX 151	IH 137	15215	17662
179: AX 146	OW 104	AE 129	IH 139	EH 131	UW 90	AE 126	7513	7652
180: IH 137	EH 131	AE 130	UW 86	UW 90	AE 129	OW 104	6898	8550
181: OW 104	AE 129	AX 146	IH 137	UW 90	AE 130	EH 131	5756	6898
182: UW 86	AX 146	IH 137	UW 90	OW 104	ER 123	AE 129	7652	7678
183: AX 146	OW 104	AE 129	IH 137	AX 149	UW 90	UW 86	6166	8021
184: AX 146	UW 90	UW 86	OW 104	IH 137	AE 129	ER 123	6955	9923
185: AX 146	OW 104	AE 129	UW 90	OW 99	AH 113	AX 149	3821	5458
186: UW 90	OW 104	AX 146	AE 129	AH 113	OW 99	AH 118	4743	5058
187: AX 146	OW 104	UW 90	AE 129	AH 113	OW 99	AH 118	4273	4320
188: UW 90	AX 146	OW 104	AE 129	AH 113	ER 122	AX 149	4224	5914
189: ER 161	AE 128	AH 113	AX 149	AX 154	UW 91	OW 104	6313	6855
190: ER 120	HH 48	V 31	AX 154	HH 46	AX 153	AX 152	7025	13314
191: ER 120	AX 152	M 54	UW 91	V 31	AX 154	HH 48	12881	17683
192: M 54	Y 165	N 63	UW 91	AX 152	NX 70	AX 147	3206	6964
193: M 54	Y 165	AX 147	UW 91	N 63	M 56	NX 69	6987	8007
194: Y 165	M 54	AX 147	UW 91	AX 151	N 63	NX 69	5986	9524
195: UW 86	ER 123	AX 150	IH 137	UW 94	AX 151	IY 143	6422	10170
196: AX 150	UW 86	IY 143	IH 137	ER 123	UW 94	IH 138	8861	9177
197: IY 143	AX 150	UW 86	UW 94	N 65	IH 137	ER 123	10383	10483
198: AX 150	UW 86	IY 143	N 65	IH 137	UW 94	ER 123	9717	10855
199: UW 86	AX 150	IY 143	ER 123	N 65	UW 94	IH 137	11045	11239
200: IY 144	N 65	AX 150	P 12	AX 151	AX 149	D 17	11088	13978
201: NX 68	M 56	N 60	NX 69	M 54	NX 71	AX 154	5191	12991
202: V 32	N 64	W 76	W 74	N 59	- 3	W 75	5832	7241
203: N 64	V 32	W 74	W 76	- 3	N 61	N 59	2503	12900
204: V 32	N 64	N 61	N 58	W 75	N 59	- 3	9646	14177
205: - 2	N 61	N 58	W 75	M 51	N 62	V 32	3266	34473
206: - 2	N 62	N 58	W 75	M 51	N 66	V 32	12574	18834
207: W 78	DH 37	V 35	P 11	N 59	L 82	L 83	380	19873
208: W 78	DH 37	V 35	P 11	L 82	L 83	N 59	2535	14098
209: - 3	W 75	N 59	W 74	N 62	V 32	W 76	7743	9985
210: V 34	HH 47	ER 125	L 81	- 4	V 31	AX 148	4227	8393
211: HH 47	ER 125	V 34	- 4	L 81	HH 45	HH 46	2035	2938
212: HH 46	ER 124	ER 120	AX 153	V 31	HH 47	AX 155	9803	10147
213: AE 128	ER 161	ER 122	DW 104	AH 113	UH 96	AH 118	6678	9896

AP News Retrieval Task:

Let me have all the stories.
Let me have all the stories.

Give me France.
Give me France.

Tell me all about Nixon.
Tell me all about Nixon.

Tell me about Watergate.
Tell me about Watergate.

Tell us all about China.
Tell us all about China.

Give us Russia.
Give us Russia.

Tell me all about Isreal.
Tell me all about Israel.

Let me have the headlines.
Let me have the headlines.

Give me the summary.
Give me the summary.

Interactive formant tracking task:

I want to do formant tracking.
I want to do formant tracking.

Use a Hamming window with five hundred, twelve points.
Use a Hanning window to five hundred, four points.

Increment the window in steps of one hundred points.
Increment the window in steps of one hundred points.

For each window, compute the fast Fourier transform.
For each window, compute the fast Fourier transform.

Display the Fourier spectrum.
Display the Fourier spectrum.

Display the LPC smoothed spectrum.
Display the LPC smoothed spectrum.

Display the cepstrally smoothed spectrum.
Display the cepstrally smoothed spectrum.

Use a pre-emphasis of six db per octave.
Use a pre-emphasis of sixty db per octave.

Medical questionaire task:

Do you smoke?
Do you smoke?

Do you drink?
Do you drink?

Do you have numbness?
Is your numbness?

Where is the pain?
Where is the pain?

Have you had mumps?
Is your numbness?

Are your headaches severe?
Are your headaches severe?

Are you in pain?
Are you in pain?

Where were you hospitalized?
Where were you hospitalized?

When were you immunized?
When were you immunized?

Have you been circumcised?
Have you been circumcised?

Is the pain severe?
Is the pain severe?

Have you ever been anesthetized?
Have you ever been anesthetized?

Have you ever been injured?
Have you ever been injured?

Have you ever had an operation?
Have you ever had an operation?

How often do you have nausea?
How often have you had an operation?

How long have you had asthma?
How long have you had asthma?

Is your dizziness continuous?
Is your dizziness continuous?

Are you afraid of surgery?
Are you afraid of surgery?

How much do you weigh?
How much do you smoke?

Is your urine cloudy?
Is your urine cloudy?

Were you ever hospitalized?
Were you ever hospitalized?

Voice chese task:

Pawn goes to king four.
Pawn goes to king four.

Knight moves to king bishop three.
Knight moves to king bishop three.

Bishop goes to bishop four.
Bishop goes to bishop four.

Knight on king bishop three goes to knight five.
Knight on king bishop three goes to king five.

Pawn captures pawn.
Pawn captures pawn.

Knight on king knight five captures pawn on king bishop seven.
Knight on king knight five captures pawn on king bishop seven.

Queen goes to bishop three.
Queen goes to bishop three.

Knight goes to bishop three.
Knight pawn goes to bishop three.

Knight captures knight on queen five.
Knight captures knight on pawn four.

King to queen one.
King to queen one.

Knight takes pawn.
Knight takes pawn.

Knight captures rook on queen rook eight.
Knight captures rook on queen rook two.

Queen goes to queen five.
Queen goes to queen five.

Pawn on queen two goes to queen four.
Pawn on queen two goes to queen four.

Bishop moves to knight five, check.
Bishop moves to knight five, check.

Bishop goes to knight five, check.
Bishop goes to knight five, check.

Queen on queen five captures queen, check.
Queen on queen one captures queen, check.

Queen moves to queen five, check.
King moves to queen five, check.

Queen takes bishop on queen six.
Queen takes bishop on queen six.

Rook moves to king one.
Rook moves to king one.

Rook moves to king seven, check.
Pawn moves to king seven, check.

Queen moves to queen bishop seven.
Queen moves to queen bishop seven.

Interactive formant tracking task:

I want to do formant tracking.
I want to do formant tracking.

Use a Hamming window of five hundred twelve points.
Use a Hamming window of five hundred points.

Use utterance number six of file number five.
Use utterance number six of file number five.

Increment the window in steps of one hundred points.
Increment the window in steps of four points.

For each window, display the Fourier spectrum.
For each window, display the formant tracks.

Compute the LPC smoothed spectrum using the autocorrelation method.
Compute the LPC smoothed spectrum using the autocorrelation method.

Compute the roots of the inverse filter using Bairstow's method.
Compute the roots of the inverse filter using Bairstow's method.

Display the imaginary part of the roots.
Display the imaginary part of the roots.

I want to compare the autocorrelation method with the covariance method.
I want to compare the autocorrelation method and the covariance method.

Increment the window by one hundred points.
Increment the window by one points.

Display the FFT spectrum.
Display the FFT spectrum.

Use a Hanning window of two hundred, fifty-six points.
Use a Hanning window of two hundred, six hertz.

Display the FFT spectrum.
Display the FFT spectrum.

Compute the Hilbert transform.
Use two points.

I want to look at image enhancement with different parameters.
I want to compare image enhancement with different parameters.

Display the spectrogram ~~with a pre-emphasis of six decibels per oct~~ave.
Display the spectrogram to a pre-emphasis of six thousand five hertz.

Use a ceiling of thirty with a floor of zero.
Use a ceiling of ten to a floor of zero.

For each utterance display the spectrogram.
For each utterance display the spectrogram.

BIBLIOGRAPHY

[A1] Alter, R., "Utilization of Contextual Constraints in Automatic Speech Recognition," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-16, 1968, pp. 6-11.

[B1] Bahl, L.R., "Overview of the IBM Speech Recognition System," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, p. 55.

[B2] Baker, J.K., "Machine-Aided Labeling of Connected Speech," In *Working Papers in Speech Recognition—II,* Computer Science Department, Carnegie-Mellon University, 1973.

[B3] Baker, J.K., "The DRAGON System—An Overview," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* Vol. ASSP-23, February, 1975, pp. 24-29.

[B4] Bakis, R., personal communication.

[B5] Barnett, J.A., "A Phonological Rule Compiler," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, pp. 188-192.

[B6] Barnett, J., "A Vocal Data Management System," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-21, 3, June, 1973.

[B7] Bates, M., "The Use of Syntax in a Speech Understanding System," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* Vol. ASSP-23, February, 1975, pp. 112-117.

[B8] Baum, L.E., "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process," *Inequalities,* Vol. III, 1972, pp. 1-8.

[B9] Bellman, R.E., *Dynamic Programming,* Princeton University Press, 1957.

[B10] Booth, T.L., "Probability Representation of Formal Languages," *IEEE Tenth Annual Symposium on Switching and Automata Theory,* November, 1969.

[B11] Bridle, J.S. "An Efficient Elastic-Template Method for Detecting Given Words in Connected Speech," British Acoustical Society "Spring Meeting", London, 1973.

[C1] Cohen, P.S., and R.L. Mercer, "The Phonological Rule Component of a Speech Recognition System," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, pp. 177-187.

[D1] Dixon, N.R., and C.C. Tappert, "Intermediate Performance Evaluation of a Multi-stage System for Automatic Recognition of Continuous Speech," IBM, for Rome Air Development Center, RADC-TR-73-16, 1973.

[E1] Ellis, C.A., "Probabilistic Languages and Automata," Rept. No. 355, Department of Computer Science, University of Illinois, October, 1969.

[E2] Erman, L.D., R.D. Fennell, V.R. Lesser, and D.R. Reddy, "System Organizations for Speech Understanding: Implications of Network and Multiprocessor Computer Architectures for AI," Proc. 3rd Inter. Joint Conf. on Artificial Intelligence, Stanford, Ca., 1973, pp. 194-199.

[F1] Fano, R.M., "A Heuristic Discussion of Probabilistic Decoding," *IEEE Trans. on Inform. Theory,* **IT-9**, pp. 64-74, 1963.

[F2] Forgie, J.W., "Overview of the Lincoln System," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, p. 27.

[F3] Fu, K.S. and T. Li, "On Stochastic Automata and Languages," *Information Sciences,* Vol. 1, pp. 403-420, 1969.

[G1] Garvin, L., and E.C. Trager, "The Conversion of Phonetic into Orthographic English: A Machine Translation Approach to the Problem," AD425819, 1963.

[G2] Grenander, U., "Syntax-Controlled Probabilities," Tech. Report, Division of Applied Mathematics, Brown University, 1967.

[H1] Huang, T. and K.S. Fu, "On Stochastic Context-free Languages," *Information Sciences*, Vol. 3, pp. 201-224, 1971.

[I1] Itakura, F., "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, February, 1975, pp. 67-71.

[J1] Jelinek, F., "A Stack Algorithm for Faster Sequential Decoding of Transmitted Information," IBM Research Report, RC-2441, April, 1969.

[J2] Jelinek, F., "A Fast Sequential Decoding Algorithm Using a Stack," *IBM Journal of Research and Development*, **13**, pp. 675-685, 1969.

[J3] Jelinek, F., L.R. Bahl, and R.L. Mercer, "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, pp. 255-260.

[K1] Klovstad, J.W., and L.F. Mondshein, "The CASPER Linguistic Analysis System," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, February, 1975, pp. 118-123.

[L1] Lea, W.A., M.F. Medress, and T.E. Skinner, "A Prosodically-Guided Speech Understanding Strategy," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, February, 1975, pp. 30-37.

[L2] Lesser, V.R., R.D. Fennell, L.D. Erman, and D.R. Reddy, "Organization of the HEARSAY II Speech Understanding System," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, February, 1975, pp. 11-23.

[L3] Lowerre, B.T., "A Comparative Performance Analysis of Speech Understanding Systems," Computer Science Department, Carnegie-Mellon University, (in preparation).

[N1] Nash-Webber, B., "Semantic Support for a Speech Understanding System," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, February, 1975, pp. 124-128.

[N2] Newell, A., J. Barnett, J. Forgie, C. Green, D. Klatt, J.C.R. Licklider, J. Munson, R. Reddy, and W. Woods, *Speech Understanding Systems: Final Report of a Study Group*, North-Holland, 1973.

[N3] Newell, A., "Speech Understanding Systems(tutorial)" invited paper at IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974.

[P1] Paul, J.E. jr., and A.S. Rabinowitz, "An Acoustically Based Continuous Speech Recognition System," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, pp. 63-67.

[P2] Paul, J.E., A.S. Rabinowitz, J.P. Riganati, V.A. Vitols, and M.L. Griffith, "Automatic Recognition of Continuous Speech: Further Development of a Hierarchial Strategy," Rockwell International Corp., RADC-TR-73-319, 1973.

[P3] Paxton, W.H., "A Best-First Parser," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, pp. 218-225.

[P4] Paxton, W.H., and A.E. Robinson, "A Parser for a Speech Understanding System," Proc. 3rd

Joint Conf. on Artificial Intelligence,Stanford, Ca., 1973, pp. 216-222.

[R1] Rabinowitz, A.S., "Phonetic to Graphemic Transformation by Use of a Stack Procedure," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, pp. 212-217.

[R2] Reddy, D.R., and A.E. Robinson, "Phoneme-to-Grapheme Translation of English," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-16, 1968, pp. 240-246.

[R3] Reddy, D.R., L.D. Erman, and R.B. Neely, "The C-MU Speech Recognition Project," Proc. IEEE System Sciences and Cybernetics Conf., Pittsburgh, Pa., 1970.

[R4] Reddy, D.R., L.D. Erman, and R.B. Neely, "A Model and a System for Machine Recognition of Speech," *IEEE Trans. Audio and Electroacoustics,* AU-21, 3, June, 1973, pp. 229-238.

[R5] Reddy, D.R., L.D. Erman, R.D. Fennell, and R.B. Neely, "The HEARSAY Speech Understanding System: An Example of the Recognition Process," Proc. 3rd Inter. Joint Conf. on Artificial Intelligence, Stanford, Ca., 1973, pp. 185-193.

[R6] Reddy, D.R., and A. Newell, "Knowledge and Its Representation in a Speech Understanding System," in L.W. Gregg(ed.) *Knowledge and Cognition,* Lawrence Erlbaum Assoc., Washington, D.C., 1974, chap. 10.

[R7] Reddy, D.R., "On the Use of Environmental, Syntactic, and Probabilistic Constraints in Vision and Speech," Computer Science Department, Stanford University, 1969.

[R8] Ritea, H.B., "A Voice-Controlled Data Management System," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, pp. 28-31.

[R9] Rovner, P., B. Nash-Webber, and W.A. Woods, "Control Concepts in a Speech Understanding System," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, February, 1975, pp. 136-139.

[S1] Salomaa, A., "Probabilistic and Weighted Grammars," *Information and Control,* Vol 15, pp. 529-544, 1969.

[S2] Santos, E.S., "Regular Probabilistic Languages," *Information and Control,* Vol. 23, pp.58-70, 1973.

[S3] Shoup, J.E., "Research on Speech Communications and Automatic Speech Recognition," Repart No. AFOSR-70-1170TR, Speech Communications Research Laboratory, Inc., Santa Barbara, Ca., 1970.

[T1] Tappert, C.C., N.R. Dixon, D.H. Beetle, Jr., and W.D. Chapman, "A Dynamic-Segment Approach to the Recognition of Continuous Speech: An Exploratory Program," IBM, for Rome Air Development Center, RADC-TR-68-177, 1968.

[T2] Tappert, C.C., and N.R. Dixon, "A Procedure for the Adaptive Control of the Interaction between Acoustic Classification and Linguistic Decoding in Automatic Recognition of Continuous Speech," Proc. 3rd Joint Conf. on Artificial Intelligence,Stanford, Ca., 1973.

[T3] Tappert, C.C., "Experiments with a Tree Search Method for Converting Noisy Phonetic Representation into Standard Othography," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, February, 1975, pp. 129-135.

[T4] Turakainen, P., "On Stochastic Languages," *Information and Control,* Vol. 12, pp. 304-313, 1968.

[V1] Viterbi, A.J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum

Decoding Algorithm," IEEE Trans. on Information Theory, Vol. IT-13, April, 1967.

[W1] Walker, D.E., "The SRI Speech Understanding System," Proc. IEEE Symposium on Speech Recognition, Pittsburgh, Pa., 1974, pp. 32-37.

[W2] Woods, W.A., and J. Makhoul, "Mechanical Inference Problems in Continuous Speech Understanding," Proc. 3rd Joint Conf. on Artificial Intelligence,Stanford, Ca., 1973, pp. 200-207.

[W3] Woods, W.A., "Motivation and Overview of BBN SPEECHLIS, An Experimental Prototype for Speech Understanding Research," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, February, 1975, pp. 2-10.