1993

Word hypothesis from undifferentiated, errorful phonetic strings

R. Thomas Sellman

Rochester Institute of Technology

School of Computer Science and Technology

Word Hypothesis From Undifferentiated, Errorful Phonetic Strings

by

R. Thomas Sellman

A thesis, submitted to The Faculty of the School of Computer Science and Technology, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Approved by : Professor John A. Biles (Chairman)

Dr. James M. Hillenbrand

Dr. Peter G. Anderson

R·I·T

Rochester Institute of Technology

Wallace Library
Post Office Box 9887
Rochester, New York 14623-0887
716-475-2562 Fax 716-475-6490

SAMPLE statements to reproduce an RIT thesis:

PERMISSION GRANTED
Title of thesis WORD HYPOTHESIS FROM UNDIFFERENTIATED, ERRORFUL PHONETIC STRINGS

I R. THOMAS SELLMAN hereby **grant** permission to the Wallace Memorial Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Date: 6/22/93 **Signature of Author:** _____

PERMISSION FROM AUTHOR REQUIRED
Title of thesis _____

I _____ **prefer to be contacted** each time a request for reproduction is made. I can be reached at the following address:

_____
_____
_____

PHONE: _____

Date: _____ **Signature of Author:** _____

PERMISSION DENIED
Title of thesis _____

I _____ hereby **deny** permission to the Wallace Memorial Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part.

Date: _____ **Signature of Author:** _____

Abstract

This thesis investigates a dynamic programming approach to word hypothesis in the context of a speaker independent, large vocabulary, continuous speech recognition system. Using a method known as Dynamic Time Warping, an undifferentiated phonetic string (one without word boundaries) is parsed to produce all possible words contained in a domain specific lexicon. Dynamic Time Warping is a common method of sequence comparison used in matching the acoustic feature vectors representing an unknown input utterance and some reference utterance. The cumulative least cost path, when compared with some threshold can be used as a decision criterion for recognition. This thesis attempts to extend the DTW technique using strings of phonetic symbols instead.

Three variables that were found to affect the parsing process include: (1) minimum distance threshold, (2) the number of word candidates accepted at any given phonetic index, and (3) the lexical search space used for reference pattern comparisons. The performance of this parser as a function of these variables is discussed. Also discussed is the performance of the parser at a variety of input error conditions.

Acknowledgements

I would like to extend my sincere thanks and appreciation to Dr. Jim Hillenbrand for coaching me thorough this long and arduous endeavor. His experience and expertise in the field of speech recognition was invaluable. I am indebted to Dr. Hillenbrand as his encouragement, assistance and expert tutelage was indispensable in keeping this thesis focused.

I must also extend many thanks to Al Biles for all his suggestions and input into the research. In a multi-functional role as RIT advisor, professor for several courses, and committee chairman, I am appreciative of his analysis and guidance.

The following people deserve thanks for their roles as part-time advisors, consultants, and reviewers: Rob Gayvert, Harvey Rhody, Betsy Richards and Tom Ridely. I would be terribly remiss if I did not include thanks to my wife Connie for her endless patience and support during the many late nights and weekends during this period.

This research was in part supported by Rome Air Development Center (Griffiss AFB, New York) and done in cooperation with the Rochester Institute of Technology Research Corporation (Rochester, New York).

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Spoken language is one of the mental faculties that most identifies us as human [WINS84], and thereby becomes a prime area of consideration for artificial intelligence research. George White identified speech communication research as a "major force" in man-machine interaction, justified by that fact that "speech remains unrivaled as the fastest and most convenient way for human beings to communicate interactively" [WHIT76]. Martin's study [MART87] of the utility of speech input in user-computer interfaces found evidence to support two claims: (1) speech provides a more efficient response channel than typed input, and (2) speech supplies an added response channel increasing user productivity, particularly in situations where parallel tasks are distributed across multiple mental resources. Martin also suggested that continued research in speech recognition technologies was warranted by the future utility of speech input.

This thesis is a part of an ongoing project at Rochester Institute of Technology Research Corporation. The project's aim is to conduct research in the development of a speaker-independent, large vocabulary, continuous speech understanding system, applying techniques of artificial intelligence in addition to other disciplines.

As will be explained in the following sections, some speech recognition methods provide a sequence of phonetic symbols that are representative of some utterance. The sequence is searched for conceptually larger forms (words) that are consistent with a dictionary (lexicon). The search involves selection of phonetic reference patterns from the lexicon, then their comparison against the phonetic representation of the input utterance. Based on a measure of similarity (to be defined in later sections), the presence of reference words in the input utterance is hypothesized or rejected. The collection of these procedures is known as the process of lexical access.

The principal goal of this thesis was to implement and investigate a method of searching a phonetic string from an unknown utterance for the occurrence of words consistent with those contained

in a domain-specific lexicon. This required the establishment of knowledge sources, determining their representations, and the provision of some search procedure to progress through the unknown utterance to produce hypotheses of words contained within the utterance. The search procedure used was based on the concept of dynamic programming, which is described as follows. When searching from an initial state to a goal state, all paths from the initial state through an intermediate state can be ignored except the least-cost path to that intermediate state. Therefore, the use of dynamic programming prunes away non-vital paths, reducing the time and computational resources for the search. Details of this search concept are presented in Chapters 2 and 3.

Chapter 2 is a discussion of issues in the field of speech understanding. Some previous systems are overviewed, followed by a presentation of the architecture of the R.I.T. Research Corporation Speech Understanding project. Approaches to string comparison are covered with an explanation of the principal methods currently in use.

Chapter 3 covers the experimental methods of the thesis project itself. Topics include the knowledge sources used and their preparation, recognition techniques, algorithm constraints and conditions, evaluation criteria, and the software/hardware tools used.

Chapter 4 presents a discussion of the results obtained during experimentation. Conclusions and suggestions for further study are found in Chapter 5.

CHAPTER 2

SPEECH UNDERSTANDING

2.1. Concepts of Speech

Speech is a process used for message transfer from one individual to another, involving the generation and reception of complex acoustical signals. The process may be thought of as a coding and decoding operation over a hierarchy of levels [HYDE72].

At the highest level we have the formation of fundamental concepts, or thoughts, which are encoded as words on the linguistic level. These words are encoded on successively lower levels, involving neural processing and articulatory movements[1], until we reach the lowest level - the acoustic signal. Speech understanding is the reverse process, trying to take the acoustic level information and work back up the hierarchy to some meaningful interpretation.

These hierarchical levels in effect describe some of the basic knowledge sources available for the task of speech understanding, including semantic, syntactic, prosodic, and acoustic knowledge sources [WHIT76, REDD76]. Semantic knowledge includes word meanings and their relationships, while syntactic knowledge refers to the structural aspects of the language (e.g. word order). Prosodic sources are speech features like stress, intonation, and rhythm. Acoustic signal characteristics like energy level, fundamental and formant frequencies, and zero-crossing rates form yet another source of knowledge. Although the speech levels and events can be described, it is often unclear how the knowledge from one level is transformed into the type of information used on a successive level. Not knowing these transformation methods is an inherent difficulty in developing an Automatic Speech Understanding System (ASU) [WHIT76]. The degree of difficulty depends on what type of speech understanding system is desired or required.

---

[1]Articulators are structures in the mouth that influence sounds by modulating the flow of air. They include the tongue, teeth, lips, and structures that form the roof of the mouth. Neural processes send messages to control other physical structures that participate in producing sound waves [WITT82].

2.2. Automatic Speech Understanding Systems

2.2.1. Difficulties in ASU

Three primary factors control the level of difficulty of the automatic speech understanding problem, and in effect categorize the types of systems that have been, and continue to be investigated [REDD76, PARS86]. The first factor concerns whether the system is designed to recognize speech produced by one, or more than one speaker (speaker dependent vs. speaker independent). Speech signals contain talker-dependent information which results in a large disparity between acoustic patterns representing the same utterance when spoken by different individuals [PARS86]. This disparity between individuals is primarily caused by differences in characteristics like vocal tract size and configuration, voice pitch, and dialect. Single speaker recognition systems do not have to account for these differences to the same degree as multiple speaker systems. The variations are minor by comparison and can be compensated for by effective training techniques.

The second factor is whether the system is intended to recognize isolated words (discrete) or continuous speech. Words in isolation provide the easiest opportunity to identify the start and end of a word in an acoustical pattern [PARS86]. When people talk in a continuous fashion, boundaries between words become blurred except in a high level cognitive sense. In addition, words are strongly influenced by, and can themselves influence surrounding words. This is due to coarticulation and phonological recoding [KLAT75, OSKI75, REDD76, SMIT80, ZUE80, WITT82]. Words in isolation tend to be pronounced more carefully, suggesting a more consistent acoustic pattern [PARS86].

The third factor concerns the issue of small or large vocabularies. Small vocabularies limit the amount of confusibility [REDD76]. As the vocabulary size increases, the degree of similarity between words grows, so the ability to distinguish small differences needs to be increased. Computational requirements of search procedures also grow as the vocabulary size expands. Finally, as the size of the vocabulary increases, considerations must be given to organizational issues. System performance degrades and storage requirements increase as vocabulary size increases [REDD76, SHIP82].

2.2.2. Extensibility

The techniques for single speaker, small vocabulary, isolated word recognition systems are well understood [ITAK75, MART75, WHIT75]. However these techniques are not likely to work well with large vocabulary, multiple speaker, continuous word understanding systems [REDD76]. Speaker dependent, isolated word recognition systems with small vocabularies use template matching methods of low-level acoustic patterns, where the unit of recognition is a word, or short utterance [REDD76, ZUE80, LEVI83, PARS86]. An unknown word is analyzed to its representative acoustic pattern, and then compared to patterns stored in a lexicon. These reference patterns are a result of the speaker repeating the vocabulary one or more times and storing the acoustic pattern, in effect training the system. It is significant to note that there is a direct mapping between the input utterance and the lexicon.

Extending the above approach to large vocabulary, multiple speaker systems presents problems. The principle problem is that the search space (using acoustical template knowledge sources) grows too large for reasonable efficiency [REDD76]. An acoustic pattern representing one word requires 560 bytes for storage [ITAK75]. A 2000 word vocabulary would require over one megabyte of storage space. As the vocabulary grows, the lexicon size becomes cumbersome. Adding duplicate patterns for all vocabulary words to account for the variability between multiple speakers would expand the search space to an unmanageable size.

Additionally, words in continuous speech are affected by the contextual influences of surrounding words, so the lexicon would have to store all potential reference patterns reflecting these influences. It would take 10 million reference patterns to account for all possible 7 digit sequences in a 10 word vocabulary [REDD76]. At 560 bytes/pattern from the earlier example, over five gigabytes of storage would be required. This is unsuitable both computationally and from a storage organization standpoint.

It has been suggested [WINS84, WHIT76] that many search problems may be reduced by using a better knowledge representation. This can be accomplished by noticing that spoken words are perceived as a succession of elemental sound units called phonemes [WATE86, SHOU80]. Categorized as vowels, liquids, glides, stops, fricatives, affricates and nasals, there are approximately 40 in the English language. The use of phonemes transforms continuous acoustic input into a discrete symbolic represen-

tation, eliminating extraneous information while preserving that which is important [HYDE72]. Since words average 10 phonemes [WHIT76] and require one byte of storage, a 2000 word lexicon could be reduced to approximately 20 Kbytes. A phonetic representation provides a knowledge source that is over an order of magnitude smaller than the acoustic representation. Also, if one designs a talker-independent phonetic analyzer, many speaker variability problems go away (not dialect problems, though).

2.2.3. Previous Systems

Speech understanding systems all use a hypothesize-and-test paradigm for the recognition problem [WHIT76, SMIT80]. The general process can be guided in either a top-down or predictive fashion (as was typical in early systems) [WOLF77, LOWE80], a bottom-up or data driven approach, or a mixture of both [LESS75]. The top-down approach uses pragmatic, semantic and syntactic knowledge sources to guide the hypothesization of words, phrases and sentences, based on task-dependent knowledge and grammatical constraints. Bottom-up or "data-driven" hypothesization attempts to deduce words based on acoustical representations in the utterance. Both predictive and data-driven methods have their drawbacks. As task domains, vocabularies and acceptable grammatical forms increase, the predictive methods can become overwhelmed with potential choices. On the other hand, if the acoustic data become more corrupt, the data-driven approach will hypothesize incorrect words without being able to recover.

Of the continuous word recognition systems developed in conjunction with the ARPA research effort [KLAT76], Harpy [LOWE80] performed best (i.e., meeting the ARPA criteria). All knowledge sources (semantic, syntactic, lexical, phonetic, and acoustic) were compiled into a 15,000-state transition network. A beam search traversed the network (bottom-up) looking for optimal sequences as compared to the utterance. A major problem with this representation was its rigidity. Changes to any knowledge source as a result of expanded constraints (changes in vocabulary size, number of speakers, task domain and grammar) required a 13 hour re-compilation of the state transition network. One should note that of the ARPA systems, Harpy had the most restrictive syntax, biasing the performance comparison.

Hearsay-II [LESS75], also developed at Carnegie-Mellon, attempted to take advantage of potential parallelism by using asynchronous, independent knowledge sources (semantic, syntactic, lexical, phonetic, and acoustic) communicating via a global data structure called a blackboard. Predictive hypotheses from semantic and syntactic constraints could be used to verify hypotheses from data-driven knowledge sources, and visa versa. At any given time a knowledge source may present information to the blackboard, providing an opportunity for other knowledge sources to be activated.

2.2.4. Current System Architecture

This thesis is part of a speaker independent, large vocabulary, continuous speech understanding system under development at R.I.T. Research Corporation. The system is primarily data-driven and is void of complex control structures such as the blackboard approach of Hearsay-II. The belief is that given accurate phonetic transcriptions, a bottom-up paradigm is sufficient. This view is shared by Rudnicky [RUDN87], who developed a word hypothesizer at Carnegie-Mellon University.

Figure 1 RITRC Speech Project Software Architecture

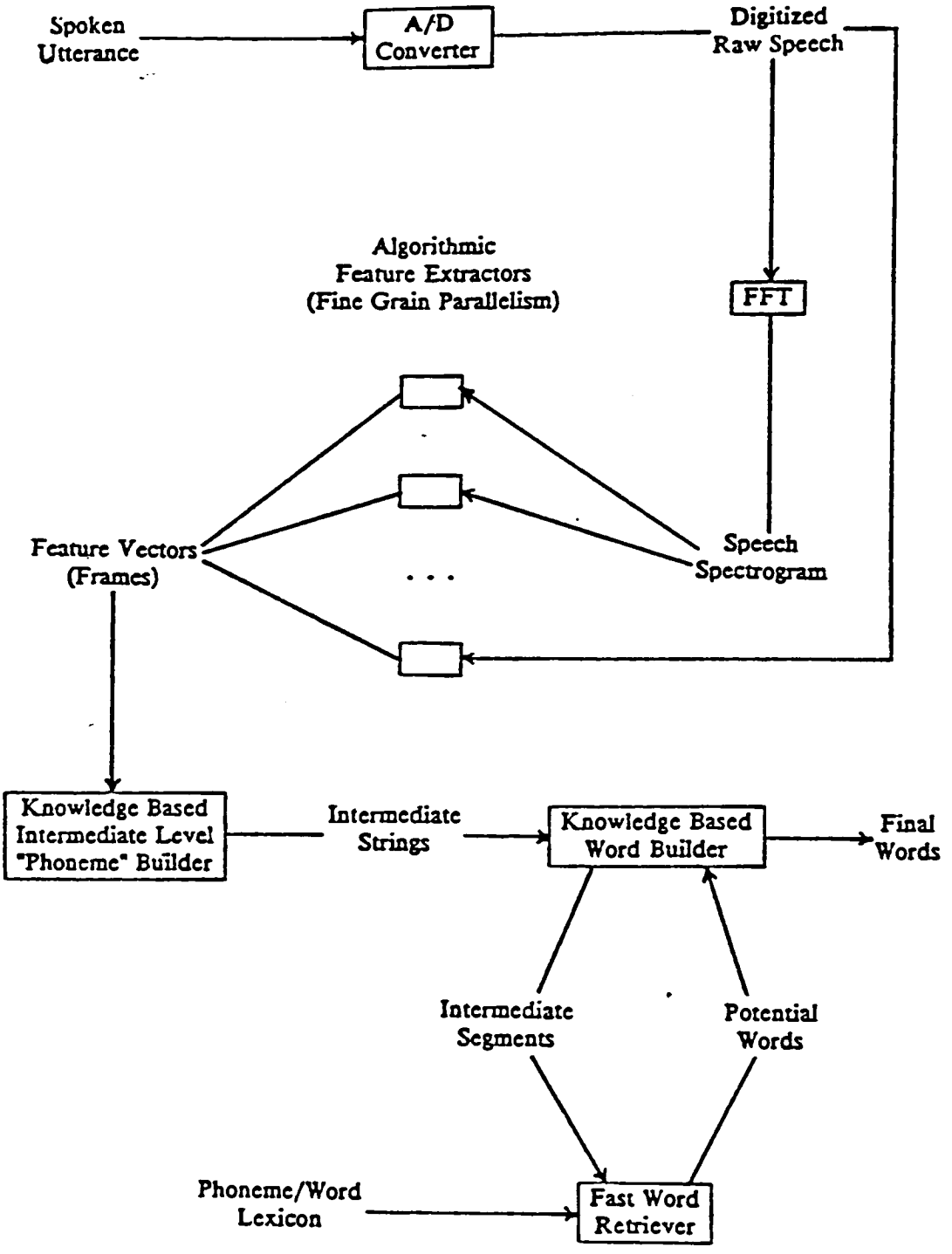


Figure 1 shows the software architecture of the entire speech project. An input utterance is processed to produce a speech spectrogram, which is a mapping of input signal frequency and intensity over time. These spectrograms preserve all speaker dependent information in addition to the phonetic characteristics of the utterance [SMIT80]. Then a set of parallel feature extractors build frames of feature vectors consisting of information such as formant frequencies, pitch, total energy and zero crossing counts. The purpose of this data compression is to preserve information that is most closely associated with phonetic content. These feature frames are subjected to a knowledge based phoneme builder to produce strings of undifferentiated phonemes (strings with no word boundary markers). Through the process of lexical access, words are hypothesized from the undifferentiated phoneme string using the string comparison technique of Dynamic Time Warping. This technique is the focus of the thesis and will be described in greater detail later. In the final stage, syntactic and semantic knowledge sources (natural language analysis, semantic networks) use the hypothesized words in an attempt to form a syntactically correct utterance and ascertain its meaning.

2.3. Lexical Access

The proposed lexical access process will parse a sequence of phonemes representing an unknown utterance, hypothesizing all words in the utterance that are consistent with the lexicon. This can be accomplished by comparing reference patterns in a phonemic lexicon with the unknown sequence.

However, there are three areas of complexity which prevent the lexical access procedure from being a simple lexicon lookup. Front end errors, and the effects of phonological recoding, are two areas which alter the symbolic representation of an utterance. Ambiguity that results in multiple parsings from a single phonetic representation is the third area.

2.3.1. Error Conditions

The front-end[2] of a speech understanding system will at times exhibit an inability to distinguish between similar sounding phonemes [PARS86]. As a result of this confusibility, the string of phonemes representing the unknown utterance will contain errors. Not only does this create a major problem when trying to match reference patterns, but any speech segment may represent three different types of errors including: insertion, deletion, or substitution errors. Finding a method that manages all three error types is not simple. Shown below are examples of the three error types.

Insertion Error - *chauffeur* : ʃ o f r → ʃ o l f r

Deletion Error - *hallway* : h ɔ l w e → h ɔ w e

Substitution Error - *tell* : t ɛ l → k ɛ l

In continuous speech, there are rule governed variations in pronunciation, especially across word boundaries [REDD76, KLAT75, OSKI75]. Figure 2a contains spectrograms of the utterance "Did you see it on the refresh screen?", spoken both in continuous speech, and as isolated words. The enlarged view in figure 2b shows the significant difference in acoustic patterns when words are spoken in isolation vs continuous speech. These variations are not random and can be described by a set of phonological rules [KLAT75, OSKI75, COHE75], following the general form: W –> X / Y_Z, meaning that W becomes X in the environment where it is preceded by Y, and followed by Z.

[2] All system components or modules from initial utterance to the knowledge based phoneme builder are known collectively as the "front end" [REDD76].

Figure 2a Spectrograms of Continuous vs Isolated Speech

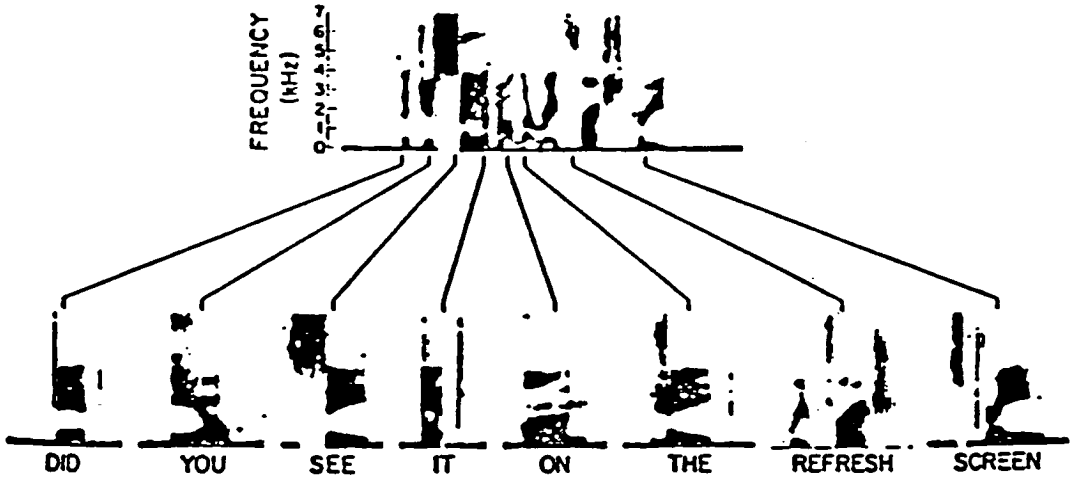
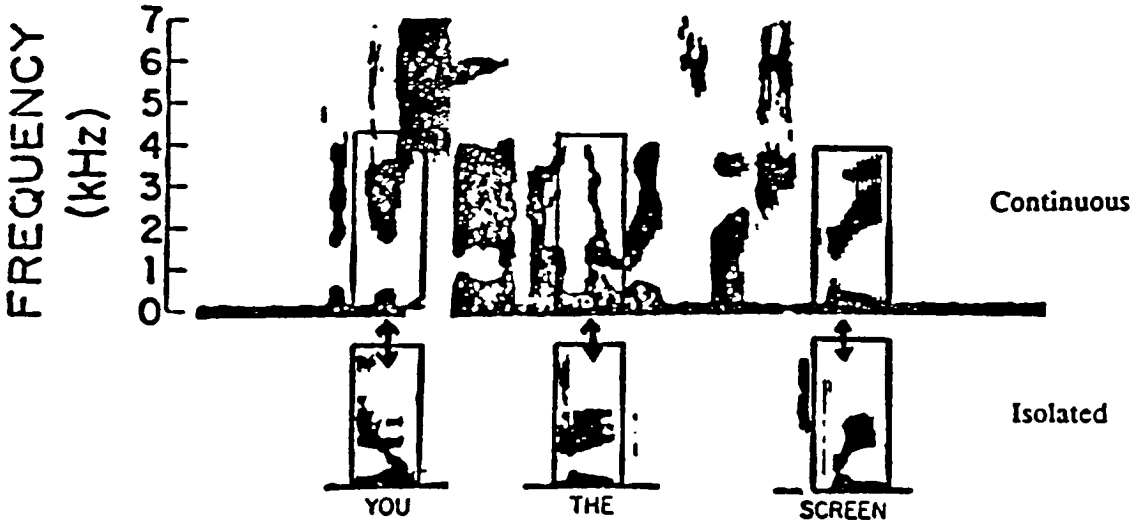


Figure 2b Continuous vs Isolated Speech (enlarged)



Finding lexical search methods that compensate for phonological recoding has not proven to be a simple matter. The following example shows the condition where a phonological rule applies within a word boundary. Specifically, the "t" is deleted when in the context of an "n" followed by an unstressed vowel. Phonological rules that apply within a word boundary can be handled by creating an alternative base forms in the lexicon.

identify → idenify

rule : t → Ø : n_V

A more difficult problem arises when working with phonological rules that apply across word boundaries. The phrase - *Did you see it?*, illustrates when an obstruent phoneme is followed by a palatal phoneme, the actual realization is reduced to a single and different palatal. Adding another baseform to the lexicon for the word "you" (starting with the palatal "j"), could provide for an erroneous recognition of "you", when the utterance may in fact be "judge".

representation when spoken in isolation : d I d j u s i t

representation when spoken continuously : d I d͡ʒ ə s i ə t

rules : [dy] → ʤ and [u,i] → ə

Even with this potential problem, many recognition systems have built lexicons where each word may have alternative representations generated by application of phonological rules to its dictionary base form representation [WOLF77, LOWE80, LESS75, RUDN87, WOOD75].

Lastly, one needs to consider the potential problem of a single phonetic string with multiple interpretations as a sentence. Ambiguous parsings can map a single phoneme sequence into different strings of words. The next two sentences have nearly identical phonetic representations, yet map onto two entirely different word sequences [COLE80].

Remember, a spoken sentence often contains many words that were not intended to be heard.

Ream ember, us poke can cent tense off in contains men knee words that were knot in tend did tube bee herd.

Matching against entries in the lexicon will not contribute to solving this problem. There is a need for syntax and semantic knowledge to differentiate meanings [REDD76]. Incorporation of syntax and domain knowledge is beyond the scope of this project.

2.3.2. Approaches To String Comparison

The two basic approaches to string comparison in isolated speech recognition are Hidden Markov Models (HMM) [LEVI83] and Dynamic Time Warping (DTW) [ITAK75]. Both methods operate on the general principle of dynamic programming (search for optimal paths) [LEVI85, KRUS83, WAIB81]. Although both approaches have been extended into the domain of continuous word recognition [WOLF77, LOWE80, LOWE80, BAKE75, MYER81, NEY84, LEVI87, WATA86], it is not certain if they are extensible to large vocabulary, speaker independent, continuous speech understanding systems.

2.3.2.1. Hidden Markov Models

Hidden Markov Models (HMM) use probabilistic techniques to model a stochastic process (Markov sources) that is not directly observable, but can be examined through a sequence of output symbols [RABI86]. Conceptually, the underlying process can be modeled with a collection of states connected by transitions; from these transitions, a finite set of symbols is produced. Through empirical testing and observation, distributions can be calculated for the probability of all state transitions, and also for the probability of symbol output given a particular transition [JELI74, JELI76, RABI86, LEVI83]. During the recognition process, a system is presented a sequence of symbols. The object whose model has the highest probability of generating the observed sequence is the one recognized.

In simplest terms, phonemes and words can be thought of as the observed output dependent on the probabilistic changes (transitions) in acoustic signals and phonemes respectively. Through the use of training utterances and empirical observation of a system's front-end performance, one can model the process of phoneme and word generation, taking into account front-end errors, speaker variation, coarticulatory and phonological recoding effects. Words within the unknown utterance would be hypothesized as those whose models had the greatest probability of generating the observed phonemes [LEVI83, BAKE33]. This statistical modeling technique was the focus of an extensive study in automatic recognition of continuous speech by Jelinek *et al* [JELI76] at IBM's Speech Processing Group.

The Dragon system by Baker [BAKE75] incorporated the concept of chaining Markov processes in a hierarchical fashion, not only at the word level, but at the phrase and sentence level as well. The result was a finite state network of Markov sources in which the recognition procedure looked for an optimal path of transitions that would most likely account for the observed utterance.

2.3.2.2. Dynamic Time Warping

Dynamic Time Warping (DTW) is a method of sequence comparison, derived from a time sampling of some quantity that is subject to variations. DTW has been successfully used in isolated [ITAK75, WAIB81] and connected word recognition [MYER81, NEY84, WATA86].

In most common applications, the unknown input utterance and reference utterance are represented as two time varying sequences of acoustic feature vectors defined [ITAK75, PARS86, WAIB81, NEY84] as:

$$\text{Unknown}: \quad A = a_1, a_2, a_3, \ldots a_i, a_M$$

$$\text{Reference}: \quad B = b_1, b_2, b_3, \ldots b_j, b_N$$

Each sequence defines the axis of a matrix mapping the feature vectors (per unit time) against one another. At each coordinate C(i,j) is a measure of distance or dissimilarity d(i,j) between the acoustic features. The goal is to find a path (with index k) from C(1,1) to C(M,N) whose distance D is minimal. This cumulative distance can then be used as a decision criterion for recognition.

$$D(A,B) = \text{Minimum} \sum_{k=1}^{K} d(i(k),j(k))$$

Ney [NEY84] applied the concepts of dynamic programming to the above minimization problem, and concluded the following:

> If the best path goes through a grid point..., then the best path includes, as a portion of it, the best partial path to the grid point...

Therefore, to obtain the best path, one only has to select the predecessor with the minimum total distance.

Following the constraints that the warping function is monotonic and continuous, a recurrence relation minimizing the number of points considered at any one time follows [KRUS83, NEY84, PARS86, SAKO78].

$$D(a_i, b_j) = \text{Min} \begin{cases} d(a_{i-1}, b_j) + w(a_i, 0) & \textit{deletion of } a_j \\ d(a_{i-1}, b_{j-1}) + w(a_i, b_j) & \textit{substitution of } a_i \textit{ by } b_j \\ d(a_i, b_{j-1}) + w(0, b_j) & \textit{insertion of } b_j \end{cases}$$

Weighting coefficients are added to penalize for deletions, substitutions and insertions. However, searching all possible paths is computationally expensive. Other constraints include controlling the degree of slope allowed in the warp, and setting some maximum permissible path distance help to control this. As shown in Figure 3 [PARS86], both constraints will prune paths that would otherwise grow excessively large.

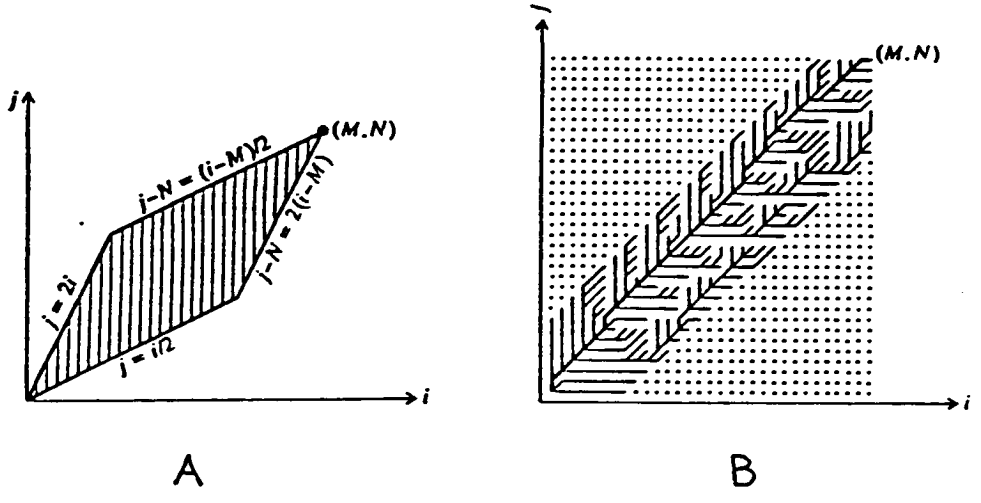


Figure 3 (A) Slope and (B) Total Distance Constraints

This thesis attempts to extend the above DTW technique using strings of phonetic symbols (phonemes) to represent the unknown and reference utterances instead of acoustic feature vectors. With this extension comes two basic differences. One is that although phoneme strings are time ordered, each symbol may represent one or more arbitrary units of time. Therefore we are not strictly warping along a time axis. The second difference relates to a central requirement of the DTW method: the need for a measure of distance or dissimilarity. When using acoustic features, these distances (commonly a simple spectral distance - e.g. distance between linear prediction coefficients [ITAK75]) are

straightforward. This is not the case when using phonetic strings. The metric that was used for phonetic string matching will be discussed in an upcoming section.

Once this measure has been determined, the process of sequence comparison can proceed. There is no requirement for any training utterances (a priori knowledge) in the recognition process as with Hidden Markov Models. Figure 4 shows two examples of the optimum path trace when using the DTW method to compare an input utterance to a reference pattern. Figure 4a demonstrates an insertion error while figure 4b shows a substitution error. The values at each coordinate indicate a measure of dissimilarity between the represented phonemes.

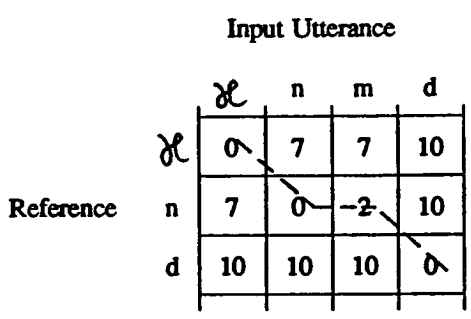


Fig. 4a

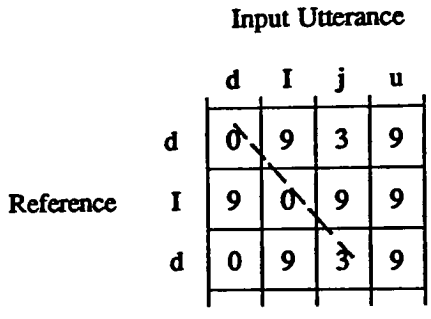


Fig. 4b

2.3.2.3. Comparison of HHM and DTW

The Hidden Markov Model is a recognition method that requires the collection of empirical statistics that describe the response of the recognition system's front-end. The determination of states, transitions, and associated probabilities is a complex optimization problem [JELI76, LEVI83]. DTW, on the other hand, needs only receive the strings for comparison and the provision of some distance metric.

The principal drawback of DTW is the large number of distance calculations that are required. It has been estimated that HMM, which uses a simpler likelihood evaluation function, requires an order of magnitude less computation time than DTW [LEVI83]. Also noted was that both systems achieved comparable error rates when applied to isolated word, template matching.

It has been noted that with a full natural language there is an infinite number of word order combinations and associated contextual influences [COHE75]. This implies that as the vocabulary size and task domain become larger, the number of HMM states needed to model multiple word forms in the lexicon grows. Since DTW methods represent each reference word with a limited number of base forms, the lexicon may grow at a more modest rate.

CHAPTER 3

EXPERIMENTAL METHOD

3.1. Software Tools

The principle part of this study was implemented in COMMON LISP, due to the predetermining fact that the project is being developed on a LISP machine. COMMON LISP is one of but many dialects of LISP (LISt Programming). LISP is a functional programming language, oriented for the manipulation symbols, and thus a favorite in AI applications. Using LISP in an interpreted fashion allowed rapid prototyping, giving the programmer quick confirmation of the success or failure of code. As a list processing language, it was well suited in manipulating sequences of tokens (phonemes). COMMON LISP was developed in an effort to combine the features of the other dialects in an optimal way, and to promote the commonality among diverging new dialects [STEE84].

Several utility programs were written in C on a Sun Microsystem workstation. The reason for this is that the initial base form phonemic representations were derived from the output of the DECtalk synthesis system which was physically separate from the LISP environment. Several small filters transformed that output into the basic LISP forms for use in the lexicon. Another program was coded to provide an interactive utility used during construction of the confusion matrix. It enabled the operator to adjust the phoneme distance matrix and view the effect on phoneme confusion probabilities as distances are varied.

3.2. Hardware Tools

The project took place on both Texas Instruments Explorer[1] I and II LISP machines. The Explorer is a microprogrammed, dedicated LISP workstation, providing a comprehensive AI environment for fast

[1]Explorer is a trademark of Texas Instruments Incorporated

symbolic processing. Additionally, the Explorer contains a TMS 32020 Signal Processor Board that allows low-level feature extraction to proceed in parallel using four independent signal processors. These characteristics allow the integration of low-level processing with the high level control mechanisms typical in AI applications. The primary difference between the Explorer I and II is that the LISP microcode on the Explorer I is distributed over a series of integrated circuits, whereas on the Explorer II it is reduced to a single chip. An increase in performance of approximately four to one was observed for roughly equivalent tests during this study.

3.3. Distance and Confusion Matrices

Implementation of the warping procedure in DTW requires some measure of distance between phonemes. Predictable confusibility patterns are exhibited by the acoustic-phonetic (i.e., *front-end*) modules of speech recognition systems. Ideally, a comprehensive inter-phoneme distance matrix would be based upon the system's front-end response characteristics in classifying all phonemes; however, vowel classification is the only portion of the front-end for which data exists. Figure 5a represents the response of the vowel classification portion of the Research Corporations front-end [HILL87] and Figure 5b shows the corresponding vowel vs vowel distances. Distance data for consonants were extracted from studies of human confusibility [SHEP80]. Studies by Miller and Nicely [MILL55] demonstrated that humans typically confuse particular consonants in a consistent fashion. In the Miller and Nicely study, listeners were asked to identify stimuli drawn from a set of 16 English consonants. The results (Figure 6a) from Shepard's [SHEP80] multidimensional scaling analysis show clusters of consonants that are similar and likely to be confused. Distances between these clusters were measured and are shown in Figure 6b.

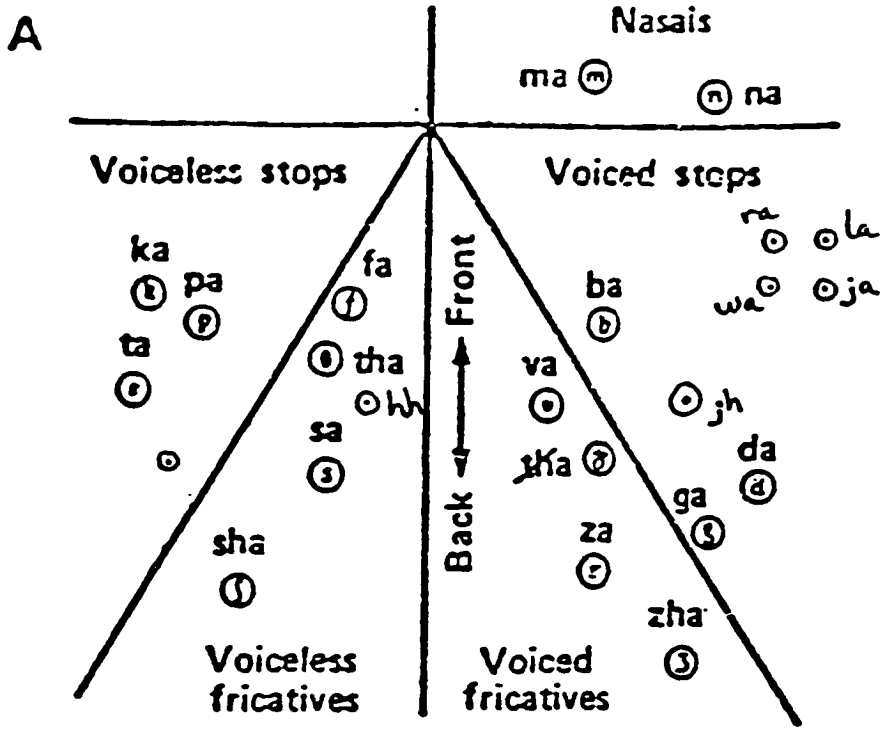
		MVD Output									
		iy	ih	eh	ae	er	ah	aa	ao	uh	uw
	iy	95.8	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	ih	7.7	85.2	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	eh	0.0	12.0	85.9	2.1	0.0	0.0	0.0	0.0	0.0	0.0
Input	ae	0.0	0.0	10.6	88.7	0.0	0.7	0.0	0.0	0.0	0.0
to	er	0.0	1.4	1.4	0.7	94.4	1.4	0.0	0.0	0.0	0.7
MVD	ah	0.0	0.0	0.0	0.0	0.7	88.7	7.7	2.8	0.0	0.0
	aa	0.0	0.0	0.0	0.0	0.0	8.5	84.5	6.3	0.7	0.0
	ao	0.0	0.0	0.0	0.0	0.0	0.7	7.0	85.9	4.2	2.1
	uh	0.0	0.0	0.0	0.0	0.7	1.4	0.0	1.4	83.8	12.7
	uw	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	14.8	84.5

Figure 5a. Vowel Classification Response - Confusion matrix showing the distribution of both correct and incorrect choices made by the MVD recognition algorithm using parameter set consisting of formant's F0, F1, F2, F3.

	iy	ih	eh	ae	er	ah	aa	ao	uh	uw
iy	0.0	7.5	22.9	62.1	57.0	88.4	121.1	112.0	76.5	85.9
ih	7.5	0.0	4.7	28.2	25.2	46.2	70.4	67.8	42.5	54.3
eh	22.9	4.7	0.0	9.8	17.0	26.3	43.3	50.1	33.6	49.9
ae	62.1	28.2	9.8	0.0	20.1	13.3	19.8	39.6	35.7	58.1
er	57.0	25.2	17.0	20.1	0.0	18.1	32.4	31.5	17.8	28.5
ah	88.4	46.2	26.3	13.3	18.1	0.0	3.0	7.4	10.2	23.5
aa	121.1	70.4	43.3	19.8	32.4	3.0	0.0	10.1	20.5	36.5
ao	112.0	67.8	50.1	39.6	31.5	7.4	10.1	0.0	5.0	10.3
uh	76.5	42.5	33.6	35.7	17.8	10.2	20.5	5.0	0.0	2.9
uw	85.9	54.3	49.9	58.1	28.5	23.5	36.5	10.3	2.9	0.0

Figure 5b. Vowel Distances - Measured spectral distances based on formant's F0, F1, F2, F3 using a maximum likelihood distance measure.

Figure 6a Shepard's Multidimensional Scaling



	k	p	t	f	th	s	sh	v	dh	z	zh	g	d	b	m	n
k																
p	8.0															
t	13.0	12.5														
f	27.0	20.0	31.0													
th	25.0	17.5	26.0	8.0												
s	33.5	26.0	28.0	22.5	15.5											
sh	41.0	35.5	29.5	40.5	32.5	19.0										
v	55.5	47.5	55.5	30.0	30.5	31.0	48.0									
dh	64.5	56.5	63.0	39.5	39.0	36.5	51.5	10.0								
z	70.0	62.0	66.0	48.5	46.0	38.5	48.0	23.0	15.0							
zh	87.0	79.0	82.5	65.5	63.0	54.5	61.0	39.0	29.5	17.5						
g	82.0	74.0	80.0	57.5	56.5	52.5	64.0	28.0	18.0	16.5	18.0					
d	86.0	78.5	85.0	60.5	61.0	58.5	71.0	31.0	22.0	25.0	25.5	9.0				
b	61.0	54.0	63.5	34.5	38.0	42.5	60.5	13.5	18.0	33.0	46.5	31.5	30.0			
m	66.5	62.0	75.0	45.0	52.5	64.5	83.5	44.5	51.5	66.0	79.0	63.0	59.5	33.5		
n	80.5	75.5	87.5	56.5	63.0	72.5	91.0	47.0	50.5	65.0	75.5	58.0	52.5	34.0	16.5	

Figure 6b. Consonant Distances - based on Shepard's MDS of Miller/Nicely's Perceptual Confusion Data

The reason for using distances from the aforementioned studies was that algorithm performance can be tested and tuned as development proceeds with the remaining front-end modules. Once all modules are complete, distance data representing actual system performance can be incorporated in the distance matrix.

One problem encountered when constructing the distance matrix was that we were missing phonetic distances: (1) relating vowels and consonants, (2) vowels not in the Research Corporation study and (3) of consonants not in the Shepard study. Based on discussions with Dr. Hillenbrand, the following assumptions were made to account for the missing data:

- Distance between vowels and consonants (with the exception of liquids and glides) was considered sufficiently large to assume a confusion probability of zero.
- Diphthongs can be thought of as transitional combinations of singular vowels (Figure 7). Instead of approximating phonetic distance over this transition, the component vowels of each diphthong (for which data exists) were substituted within the transcription of a word each time they occurred.
- Syllabic resonants were treated similarly to diphthongs. Each occurrence within a word was substituted with a similar consonant counterpart (Figure 7).
- Distances for singular vowels not in the Research Corporation's study was provided using appropriate adjustments to existing data for similar vowels (Figure 7).
- Singular consonants not in Shepard's study include liquids, glides, flaps and affricates. These were added to suggested locations [HILL88] in Figure 6a, and their distances approximated.
- Liquids and glides were unique in that they mapped to both consonants and vowels (Figure 7). Their positions relative to other consonants in Figure 6a, and their similarity to specific vowels, influenced the distances that were approximated.

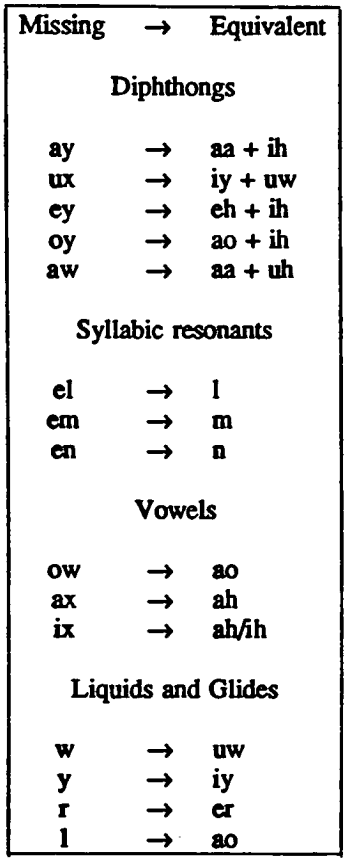


Figure 7. Mapping of Missing Phonemes to Similar Phonemes based on Perceptual Confusion

In addition to simulating the front-end, testing requires that errors be simulated in a way that approximates the front-end response characteristics from which the inter-phoneme distance matrix is derived. Since most of the phoneme distance data is based on perceptual confusion between phonemes, there is a need for probability data reflecting how often an input phoneme is confused with zero or more phonemes, producing insertion, deletion and substitution errors. Vowel confusion probabilities were obtained directly from the vowel classification study [HILL87]. For consonants, however, only their perceptual distance existed. If a relation between distance and confusion probability could be found to approximate the vowel study results, this could be applied to the consonant distances from Shepard's analysis [SHEP80] of Miller and Nicely's consonant confusion data, yielding approximate confusion probabilities.

Examination of confusion probabilities and distances from the vowel classification study yielded an approximation to the following exponential relationship ($Base \cong 1.50$):

$$\text{Confusion Probability (input vs output)} = \frac{\frac{1}{base^{distance}}}{\sum_{all\ vowels} \frac{1}{base^{distance}}}$$

Applying this formulation to all phonemes, it was discovered that the overall probability of confusion between phonemes was too low and did not reflect what could be expected in reality [HILL88]. Based on discussions with Dr. Hillenbrand and Robert Gayvert [HILL88] the formula relating phonetic distance to confusibility was modified as follows:

$$\text{Confusion Probability (input vs output)} = \frac{\frac{1}{distance}}{\sum_{all\ phones} \frac{1}{distance}}$$

An iterative process of adjusting inter-phoneme distances and recalculating the confusion probabilities led to the creation of a second phoneme versus phoneme matrix. The confusion probabilities in this matrix were then used in error generation during test data creation. The final distance and confusion matrices can be found in the Appendix A and Appendix B, respectively.

3.4. Lexicon Construction

The vocabulary for this study was taken from a United States Air Force Cockpit Natural Language study [LIZZ87]. The study provides a vocabulary of 656 words, their frequency of occurrence, and the number of times a word is preceded or followed by other words. This information may be valuable in determining the types of contextual effects to expect.

All words from the Air Force study were input into a text-to-speech synthesis system (DECtalk[2]). Output from this system was in the form of a synthesized utterance and phonetic transcription of each word using Digital Equipment Corporation's symbol set. Words whose auditory output did not accu-

[2]DECtalk is a trademark of Digital Equipment Corporation

rately reflect a generally accepted pronunciation were corrected. The entire transcription output was reviewed for correctness, with special attention to those words that were pronounced incorrectly. The transcriptions were then converted to the Carnegie-Mellon University phonetic symbol set which is used in conjunction with other ongoing projects at the RIT Research Corporation. Appendix C shows a comparison of the symbols used in both the DEC and CMU symbol sets.

For each word transcribed, a lexicon entry was constructed containing the transcription length, the transcription, and the English representation of the transcribed word. These entries (see Appendix E) are grouped based on word-initial phoneme. Placement within the group is in descending order of phoneme count, yielding the longest reference pattern first when any particular group is accessed during the search procedure. The significance of this will be explained later. Homonyms form a single lexical entry with multiple English representations. Words with multiple, but generally accepted pronunciations (e.g. hostile: *hh aa s t ay l* vs *hh aa s t el*), are given a lexical entry for each pronunciation.

3.5. Test Data Creation

The Air Force Cockpit Natural Language study [LIZZ87] served as the source of test utterances to use as input strings for the DTW process. A set of 42 test phrases was selected from the study that combined a wide variety of words available from the lexicon. The average length of phonetic transcription over the 42 phrases was 26. Test phrases were translated first to their phonemic representations with no errors, allowing some benchmark performance levels to be determined. After this, an increasingly large percentage of errors was induced into the input strings. It is important to note that errors were simulated in a manner that accurately reflected the simulated front-end response characteristics. Therefore, a key component in the error generation process was the phoneme confusion matrix. Using this matrix, the following method would generate the three different errors types, at some user defined percentage level (for each type), and at some total error rate.

While advancing through a phonetic input string, and based on the total error rate, a random number generator selected the type of error (substitution, deletion, or insertion) to occur at a given phoneme in the string. If a *no-error* condition is selected for the particular input phoneme, it maps

one-to-one into the output sequence. A deletion error results in the phoneme at that current location being dropped from the output string. If a substitution error is selected, the phoneme at the current input string location indexes into the confusion matrix, and based on the confusion probabilities for that phoneme, another phoneme is chosen for inclusion into the output string. Generation of insertion errors would also use the error matrix. The input phoneme would index into the matrix, and based on it's confusion probabilities, a phoneme would be selected for insertion into the output phoneme string.

Error types were assumed independent since it is not clear how, or if these error types are interrelated. Studies at IBM [JELI76] showed that in most cases, substitutions accounted for the majority (80% to 90%) of errors, followed by deletions and insertions. Therefore, this study concentrates on the success of the word hypothesis process using substitution errors primarily. Appendix D contains the set of test 42 test phrases with 10% substitution errors induced.

3.6. Dynamic Time Warping Process

3.6.1. Constraints and Considerations for DTW

This study used the Dynamic Time Warping technique for the hypothesis procedure. As previously described, DTW provides a method to model a warping function that maps two speech patterns onto one another. This mapping or alignment is considered optimal when the function reaches a minimum value. This function value can be used as a basis for recognition. Word hypothesis involves comparing phonemic representations of words in a lexicon to the phonemic representation of an unknown utterance, looking for those words that have an optimal alignment (minimum warping function) in consecutive time intervals not to exceed some given threshold.

A major factor that affects system performance during DTW is the value of the minimum distance threshold. Too low a value reduces the tolerance for errors and results in the premature rejection of a potential reference-to-unknown match. Too high a value will result in the acceptance of incorrect matches and the increased consumption of computational resources. A benchmark threshold value must first be established with error-free input patterns. Note that when comparing an error-free unknown to a reference pattern, any increase over zero in the accumulated distance indicates a difference in alignment

- a basis for rejecting the reference. This establishes an initial threshold for testing purposes. As the threshold for rejection is raised, it is expected that beyond some limit, the ratio of incorrect hypotheses to correct ones will increase. An optimum value for error-free patterns provides a starting point from which to investigate the effect of a minimum distance threshold on DTW performance (ratio of incorrect hypotheses to correct hypotheses, percentage of correct hypotheses) when used with error-full input patterns.

Sakoe and Chiba describe five general conditions that typically restrict the warping function [SAKO78]. The first two are that the function be monotonic and continuous. Phonemes in the reference and unknown patterns are assumed to be time-ordered with their intervals relatively uniform, satisfying the first two conditions. The three remaining conditions (established endpoints, adjustment window, and slope constraint) are variable and can affect the relative performance of the warping procedure.

Sequence endpoints are fully known for both the reference and unknown patterns in isolated word recognition. However, in continuous speech, endpoints (at the word level) in the unknown utterance are not fully established, and can be highly variable in number and position. Therefore criteria must be established for selecting the appropriate length of the unknown sequence for DTW comparison. Assuming the front-end's performance is not totally corrupt, one can expect that there is a maximum number of phonemes (including insertion, deletion, and substitution errors) in the unknown pattern which must be examined in order to find a word, or exhaust all possibilities. This value would be equal to the phoneme count of the reference word, plus a buffer to allow for insertion errors that can extend the unknown sequence. For this study, the assumption was made that no more than 100% errors were expected. The buffer value would then be equivalent to the number of phonemes in the reference pattern.

The adjustment window and slope constraint conditions affect the manner in which the DTW procedure deals with insertion and deletion errors. When finding a least cost path through the distance matrix, the warping path will cut a diagonal line with a slope of one if both patterns are aligned. The further this path deviates from the diagonal, the larger the difference between the two patterns. The adjustment window as defined by Sakoe and Chiba [SAKO78] is an area in the matrix bounded such

that the absolute difference between the indexes of both patterns is less than or equal to some constant value. This area is a diagonal corridor somewhat parallel to the warping function. This *window* constant has the effect of limiting the number of acceptable insertion and deletion errors. Excessively long horizontal or vertical paths indicate that unusual expansion or compression is required to match two patterns - an indication of poor correspondence.

Kruskal and Sankoff [KRUS83a], Myers et al. [MYER81], and Sakoe et al. [SAKO78] use the common concept of a slope constraint to limit the number of consecutive insertion or deletion errors. This results in a parallelogram that defines limits to the warping path direction (Sec.3.2.2 - figure 3). Sakoe and Chiba's [SAKO78] study defined a measure of slope (P) as the maximum number of horizontal or vertical steps (m) that could be taken before some number (n) of diagonal steps. Their study showed that optimum DTW performance maximized at P = 1 in a range from 0.5 to 2. Therefore, a slope value of one is used in this study.

Associated with the DTW equation presented in Sec.2.3.2.2 was a weighting coefficient. This coefficient allows one to apply an additional reward or penalty to the accumulated distance, accounting for path deviations. Kruskal and Sankoff [KRUS83a] illustrate the use of positive weights as a measure of quality to be included into the DTW equation that penalize for insertion, deletion, and substitution errors. For example, any movement in a horizontal or vertical direction (insertion and deletion errors) results in a positive value being added to the accumulated distance, indicating a decrease in the quality of the string match. The same is true if movement is in a diagonal direction without both string elements matching (substitution error). A diagonal move with matching string elements receives no penalty. There was not a good understanding of how to set arbitrary weighting factors of phonetic distance in response to errors. With this in mind, and in consideration of the many other variables within the DTW process, a weighting coefficient was not used in this context. However, averaging the total accumulated distance over the reference pattern length could be used to normalize distances between hypotheses whose reference patterns differ in length. This type of weighting favors a heuristic that looks for the longest pattern with minimum distance. The DTW matrix indices at the current point of comparison can be used as a divisor to average out the length traveled.

Using Sakoe and Chiba's [SAKO78] symmetric DTW equation of slope P = 1 results in the following recurrence relation to be used in calculating minimum distances during the DTW process:

$$D(a_i, b_j) = \text{Min} \begin{vmatrix} d(a_{i-1}, b_{j-2}) + d(a_i, b_{j-1}) + d(a_i, b_j) \\ d(a_{i-1}, b_{j-1}) + d(a_i, b_j) \\ d(a_{i-2}, b_{j-1}) + d(a_{i-1}, b_j) + d(a_i, b_j) \end{vmatrix}$$

3.7. Recognition Technique

3.7.1. Sentence Parsing

Conceptually, the DTW procedure moves from left to right, processing sections of the unknown sequence. A phoneme reference pattern is selected from the lexicon (its selection method will be discussed below) and time-warped with an initial portion of the unknown utterance, producing a time-normalized distance. This procedure is applied repeatedly to the same section of the unknown, until all acceptable word hypotheses are determined. Hypotheses that exceed a preset minimum distance threshold during DTW calculations are pruned early. Hypotheses that do not exceed the threshold are placed into an array at an index corresponding to the position of their word-initial phoneme in the unknown phonetic string. This array forms a word *lattice*.

For each hypothesized word (in the set generated from the initial unknown sequence), another segment of the unknown utterance is selected (*left to right*) for DTW comparison against reference patterns in the lexicon. The starting point of each "new" unknown sequence portion is taken from just after the last point of comparison between the previous unknown sequence, and the reference pattern of the word hypothesized. However this is only adequate for testing against substitution errors. Insertion and deletion errors can affect the location of the word junctures. Consider the following two unknown sequences and their representations:

(1) this may → dh ih s / *m* ey

(2) this set → dh ih s / s eh t

Example one does not present a problem. The hypothesis of *this* is made, and DTW would resume at

the phoneme "m". However, in example two the front-end would merge the "s" from both words (creating a deletion error), resulting in the sequence:

dh ih s eh t

Now when *this* is hypothesized, the algorithm advances past the phonemic representation of *this* in the unknown string to begin DTW again. The new comparison starts at the phoneme "eh", providing a more ambiguous, if not incorrect point to start the DTW procedure from. Therefore, successive DTW procedures can begin from some number of phonemes prior to the point corresponding to where each successful hypothesis ends. Although this requires additional computation, it may prevent ignoring a significant starting point for DTW.

This process continues until there is zero or more word sequences hypothesized from the unknown utterance. The lattice of hypothesized words (rank ordered based on length of phonetic transcription, then total overall accumulated distance), is sent to a syntactic and semantic parser for further processing.

The above procedure is based on the level building algorithms developed for use in connected-word recognition [MYER81, NEY84, SAKO84]. They also move from left to right through the unknown utterance, finding the collection of reference patterns whose global (phrase) DTW distance is at a minimum over the concatenation of local (word) DTW minima. Note that, given an utterance of fixed length, and given an equivalent distance between all reference and unknown patterns, a small number of large words will have less total accumulated distance (globally) than a larger number of small words, indicating a possible heuristic that favors use of large reference patterns for DTW prior to smaller patterns. Smith [SMIT80] also suggested that large words should be hypothesized prior to smaller words since larger words usually contain more syntactic and semantic value. Ordering the lexicon entries by descending transcription length is another way of exploiting this heuristic.

The top level algorithm for the word hypothesis procedure (*parse-sentence*) is as follows:

```
(DEFUN parse-sentence (input-phrase phonetic-index threshold )

   IF any more of the input-phrase to process
      {
      IF word-lattice [phonetic-index] contains hypotheses
         {
         complete = true
         }
      }

   LOOP until complete
      {
      IF at end of the input-phrase
         {
         complete = true
         }
      ELSE
         {
         hypotheses = find-eval-candidates (input-phrase phonetic-index threshold)
         }

      IF complete and no hypotheses found
         {
         (RETURN null)
         }

      IF no hypotheses are found
         {
         advance to next position in input-phrase
         IF we have advanced beyond a specified point
            {
            increment the threshold
            return to position in input-phrase where last successful search ended
            }
         }
      ELSE
         {
         word-lattice [ phonetic-index ] = hypotheses
         LOOP for all hypotheses
           {
           parse-sentence (input-phrase (phonetic-index + candidate-length) threshold )
           }
           complete = true
         }
       }

   (RETURN  word_lattice)
```

The primary function of parse-sentence is in guiding the left to right motion searching for hypotheses. Due to the recursive nature of *parse-sentence*, a previous iteration may have already found hypotheses

to exist at a particular index. Therefore, when entering the function, one must examine the word lattice at the starting index. If hypotheses already exist in the lattice at that index, there is no need to search from that index again. Otherwise, it proceeds to find all candidates in the unknown at the current phonetic index. If none are found, the index is incremented and the search process is repeated. Multiple advances without finding any candidates will eventually cause the process to back up, dynamically increase the threshold, and repeat the search. Raising the threshold provides an increased possibility of finding hypotheses by reducing the chance that they will be pruned during the DTW process. Once a set of hypotheses is located, the entire procedure is repeated using the endpoint of each hypothesized word as the next position from which to begin the search.

The function **find-eval-candidates** has two responsibilities. The first is to select reference patterns from the lexicon and initiate the time-warping process to compare each against the unknown utterance. Exhaustive search of all lexical entries is not practical with lexicons numbering in the tens of thousands. However, in order to minimize the number of factors which would influence the DTW process, complex search strategies were not investigated. The search strategy used developed from a brute force method to one in which subsets of the lexicon were selected and then compared to the unknown via DTW. This method gathers reference candidates based on the word-initial phoneme of the unknown utterance segment to be warped. As described earlier, all words in the lexicon are grouped according to word-initial phoneme. During recognition, the first phoneme in the unknown pattern is used as a key into a similarity table. In the table at each key is a list of three to five phonemes. These phonemes have the highest probability of being confused with the key. This association list can be recursively processed to provide a *family* of phonemes that are most highly confused with the initial index. How the size of this *confusion-family* relates to recognition rate and performance is one of the major variables evaluated in this study. It is important to note that this method of obtaining reference patterns relies on the premise that the first phoneme in the unknown sequence can be identified accurately.

When a reference pattern successfully completes the warping process without exceeding the distance threshold, the following information is returned and eventually placed in the word lattice:

(reference_word warping_distance reference_pattern_length next_search_index)

The reference word is the basic data component of the word lattice expected as input by upper level semantic/syntactic parsers. Warping distance serves as a measure of confidence in the accuracy of hypothesis for the reference word, and is used in a final selection process detailed next. The length of the reference pattern provides an evaluative measure used in the selection process as well. Both warping distance and reference pattern length are measures that may be beneficial to the the upper level parsers in guiding their syntactic and semantic parsing procedures. The fourth value represents the phonetic index within the unknown phrase where the next search process is to begin.

Find-eval-candidates second task is to choose a subset of hypotheses from those generated above which have the greatest confidence measure, based on minimum phonetic distance and phonetic representation length. Selection is accomplished by performing two sorts on the hypotheses found at a given index within the unknown phoneme string. The first sort rank orders hypotheses by increasing minimum distances. All but the best (lowest minimum distance) N hypotheses are discarded. Those that remain are sorted again based on decreasing phonetic representation length. This final list of hypotheses is placed in the word lattice (at the given phonetic index) allowing access to the longest hypothesis first. The parsing process as outlined above would use these hypotheses as starting points for continued analysis. This is consistent with the previous suggestion by Smith [SMIT80] that large words be hypothesized prior to smaller words since larger words usually contain more syntactic and semantic value. Find-eval-candidates outlines as follows:

```
(DEFUN find-eval-candidates (input-phrase current-index threshold )

  ref-words = get-candidates-based-on-similar-phoneme ( phoneme-at-current-index )

  (LOOP for all ref-words
     {
     get section of unknown phrase based on reference pattern length
     hypotheses = process-DTW ( reference-word unknown-phrase-section threshold )

     sort-on-DTW-distance ( hypotheses )
     trim-list-of-candidate ( hypotheses )
     sort-on-reference-pattern-length ( hypotheses )
     }

  (RETURN  hypotheses )
```

CHAPTER 4

RESULTS

4. Initial Findings

Initial tests of the DTW process were conducted using phrases with no errors and a minimum distance threshold of zero. These tests used a brute force search through the lexicon, comparing all words against the unknown sequence. Although all phrases were parsed successfully, it became apparent that the DTW process was computationally expensive. Processing time for an unknown sequence (average length of 20 phonemes) was approximately ten seconds per sequence. A design deficiency in the parsing procedure was discovered and corrected. Within recursive calls to the function *parse-sentence*, alternative parses were duplicating effort in searching identical sections of the sequence. To eliminate this, a global array was implemented to store the results from searching specific segments of the unknown at a given index. If an additional parse were to begin at the same index, previously found hypotheses (if any) would be immediately available. With this change, processing times for errorless strings were reduced by approximately 50%.

Pilot results with errorful strings showed even further degradation in efficiency. Processing of errorful strings requires that the minimum distance threshold be set to some positive value to accommodate minor mismatches in the reference-to-unknown alignment due to the three error types. As that minimum value was increased, the DTW process had more opportunity to progress through the string before exceeding the threshold, adding computations. In addition, an increased threshold also allowed more hypotheses to be found, which in turn increased the number of endpoints that would serve as new positions for further search.

In the context of only substitution errors, initial tests performed so poorly that the entire sequence of words was not found in any of 51 unknown phrase tests. Of all words hypothesized, only 46% of those found were the original words. Observations made from the examination of these trials are dis-

cussed in the following paragraphs.

It was noted earlier that searching the entire lexicon posed performance problems. In an attempt to reduce the computational overhead, the strategy for selecting reference patterns for DTW comparison was altered from the brute force approach. The procedure was changed so that the unknown sequence was compared with those reference patterns starting with either the initial phoneme of the unknown, or any of the three phonemes that had the highest degree of confusion with the initial phoneme. This reduction in calculations improved overall efficiency, but was too selective. More often than not, the reference pattern required to match the unknown would not even be selected for comparison. It became apparent that some larger segment of the lexicon is necessary for reference selection. The effect of lexical search space on DTW performance was one of the three major variables used in the series of tests discussed in the next section.

Another observation made during pilot testing was that different criteria (weighting factors) were necessary when warping with short (i.e., three or less phonemes) reference patterns as opposed to larger ones. Initially the same weighting scheme was used for all words when calculating the average distance for comparison against the threshold. It became difficult, however, to find a threshold value that would be low enough to screen against warping differences early into the comparison, yet not be so sensitive as to prune the search when encountering moderate differences. In the case of short reference words, a decision to prune the search in the event of any deviation between reference and unknown had to be made quickly. On the other hand, it was desirable to allow longer patterns to continue warping even in the event of an error. For example, the warping process proceeds with an identical match between reference and unknown until the very last phoneme which contains a substitution error of significant magnitude. Although the difference may be large, the majority of the reference pattern has been accepted by the warping process and is most likely a good hypothesis. This large distance is probably a spurious front-end error that should be discounted. Therefore, examining the effect of a range of minimum distance thresholds on DTW performance was selected as a second major variable to test.

A third observation was that many times the parsing procedure was not able to locate any hypotheses at a given index within the unknown. In this case, it advanced to the next position and

started the search again. This occurred several times, skipping over significant sections. Applying the following heuristic to the parsing process improved results. If the recognition process continues too far without finding any hypotheses, it backs up some distance within the utterance and searches again, dynamically increasing the distance threshold. This raises the likelihood of finding words at any given index. The third variable examined in the test series was the number of acceptable word candidates allowed at any given phonetic index within the utterance. Intuitively, as the number of hypotheses increases, so does the probability that the correct one is included.

Insertion and deletion errors introduced significant problems in the word recognition process. One of the most noticeable was that they could change the starting phoneme of a word. Since this study based reference pattern selection on the word-initial phoneme of the unknown sequence, deletions or insertions at word boundaries had a negative impact on successful hypothesis. Selection of reference patterns based on either the second or third phoneme of a word (in the case of deletion errors), or an inserted phoneme, would be from areas in the lexicon that did not include the required reference pattern for successful match. Another difficult problem was encountered when working with deletion errors in the case of small words (two phonemes). Accurate hypothesis from a single phoneme without setting thresholds to such a level as to cause massive acceptance of more *false-positives* is a problem.

Problems with the DTW process were not as severe if deletion and insertion errors occurred within word boundaries, but in comparison, deletion errors proved more difficult to account for than insertion errors. To explain this, one must look at how these errors are generated in the test utterances. Deletion errors are introduced randomly into the test utterances, as opposed to insertion errors which are based on data from the confusion matrix. As a result, the insertion error will create an additional phoneme, but of similar phonetic classification. In contrast, a deletion error can cause a sharp change in phonemic character not previously present. An example is when one of two vowel sounds surrounded by consonants is deleted. When warping the reference against this unknown, the consonant is now encountered prior to normal causing the average distance to exceed the threshold. As was previously pointed out, an IBM study [JELI76] showed that of the three error types, substitutions accounted for the majority. Given the above problems, and the results of the IBM study, this project worked primarily with substitution errors.

4.1. Test Series 1

A comprehensive series of tests was run against a group of 42 phrases containing 10% substitution errors. Each successive test varied one of three primary variables (holding the other two constant) found to affect performance of the word hypothesis process. These variables included: the minimum distance threshold, the number of candidate words accepted at any one phonetic index, and the size of the lexical search space when obtaining reference patterns for the DTW process. Five threshold values were used ranging from zero to 1000. Values ranging from 5 to 30 were used as the number of candidates accepted at any index. Two possible values, *large* and *small*, were tested for lexical search space size. A small search space provided access to approximately 20% to 30% of the word-initial phoneme groups in the lexicon, whereas a large search space was approximately double that of the small space. Results of the Series 1 tests at threshold values of 500 and 1000 are for only the small lexical search space. Initial trends in parallel tests indicated that differences in search space did not have a significant impact on total recognition. As will be detailed later, this observation proved to be inaccurate.

The performance of each test was evaluated using the following criteria. First, the final word lattice returned from each parsing was examined for the presence of the intended utterance. Finding all utterance words in their correct order was considered a *complete match*. The number of complete matches from N test utterances provided the total percent recognition. A second measure of success was the *average percentage of words hypothesized per phrase*. This gives a relative idea of how well the parsing process is working on a phrase basis. It relates the number of correct words found to the number in the originals in the utterance over all phrases. The last two performance measures are used as indicators of *noise* in the word hypothesis procedure. The process of word hypothesis attempts to interpret speech information passed up from the lower levels of a speech recognition systems front-end. Ideally, no information should be lost in the transfer of this information to higher levels. It is therefore desirable that as errors do occur, they should be of the *false-positive* type (words found but not present) rather than errors of omission. The two noise measures are: (1) the ratio of correct to incorrect words found, and (2) the ratio of total words found vs total words in the utterance.

From a computational perspective, the run-time performance of the Series 1 tests was much lower than expected. On an Explorer I, times ranged from 4 to 21 hours per test. Equivalent tests on the Explorer II yielded an approximately five-fold reduction in computation time. There was a positive correlation between increasing run times over the series of tests and increases in all three variables. Larger threshold values allowed the warping process to further continue comparisons of reference patterns to unknowns before exceeding the minimum threshold levels that result in process termination. Increasing the number of word candidates allowed per index produced additional starting points for new searches which also use more computational resources. Lastly, widening the search space served to increase the number of reference patterns available for DTW comparison, requiring even more DTW comparisons.

Figure 8 shows the total percent recognition achieved as a function of the DTW threshold used. In general, as the threshold was raised, the percentage of complete phrase recognition also increased. Raising the threshold causes the DTW comparison to be less restrictive and thus, provides an increased likelihood that the DTW process would complete and provide a hypothesis. A maximum recognition level of 66% occurred at threshold levels of 275, 500 and 1000. Correspondingly, both the number of accepted candidates per index, and the lexical search space variables were at their greatest values (thirty, *large*, respectively) when this maximum recognition level was achieved. As indicated by the plateau in total recognition rate, it did not look as if further increases in threshold would yield improved results.

Also shown are the high and low percentage recognition levels obtained for each threshold value. The variances in recognition reflect the effect of underlying changes in the other two variables (i.e., candidates accepted per phonetic index and size of lexical search space). With the exception of the zero threshold level test, the range of recognition levels for a particular threshold class was relatively close at five percent. This indicated that although some improvement in recognition is possible by adjusting the candidate per index and search space variables, the increase appears to be modest. The wide fluctuation in the zero level threshold test can be attributed to the heuristic that causes the parser to backup and increase the threshold in the event of multiple advances without success. As the threshold increases (in increments smaller than the differential in test category levels), it becomes large enough

for discovery of more hypotheses than at the zero threshold, but not in quantities equivalent to those tests starting with a large value initially.

Figure 9 shows how the percentage of complete recognition was related to the number of candidate hypotheses allowed at a given index in the unknown utterance. Figure 10 shows how the average percentage hypothesized per phrase was related to the number of candidate hypotheses allowed at a given index in the unknown utterance. Increases in complete recognition of two to three percent were realized as the number of candidates increased. Individual phrase recognition percentages for the Series 1 tests fluctuated between the range of 85% and 90%. As was true for total recognition figures, the average percentage of a single phrase rose two to three percent as the candidate count was increased. However, beyond 20 candidates per index, further increases appeared not to be beneficial. This indicates that some other limiting factor must exist which affects recognition capability. Also illustrated in these graphs is a clustering effect for results with thresholds at the 0 to 175 level, and those at the 225 to 1000 level. The cause for this is unknown.

Measures of noise in the Series 1 tests are shown in Figures 11 and 12. As expected, increasing the threshold level for a given test resulted in larger noise levels during word hypothesis. When the distance threshold was raised, more candidate words were able to pass this minimum difference and were therefore accepted. Tests with threshold levels under 500 demonstrated that the hypothesis of between six and 32 words was necessary to find an original utterance word. In contrast, as the number of word candidates accepted per index approached 20, the noise ratios began to stabilize, still increasing but at a decreasing rate. As the number of allowed words per index increased, so did the chance that the actual word desired would be present. Keeping in mind that the candidates allowed per index were specifically sorted, the manner in which the sort was conducted has the potential to substantially affect success of hypothesis. Tests at threshold levels above 500 produced much higher noise levels. Tests at these levels required the hypothesis of between two and three times as many words for every correct word, as did tests using lower thresholds. Tests providing the largest percentage of complete phrase recognition (55% to 66%) resulted in ratios of total words found vs total utterance words, as high as 74:1.

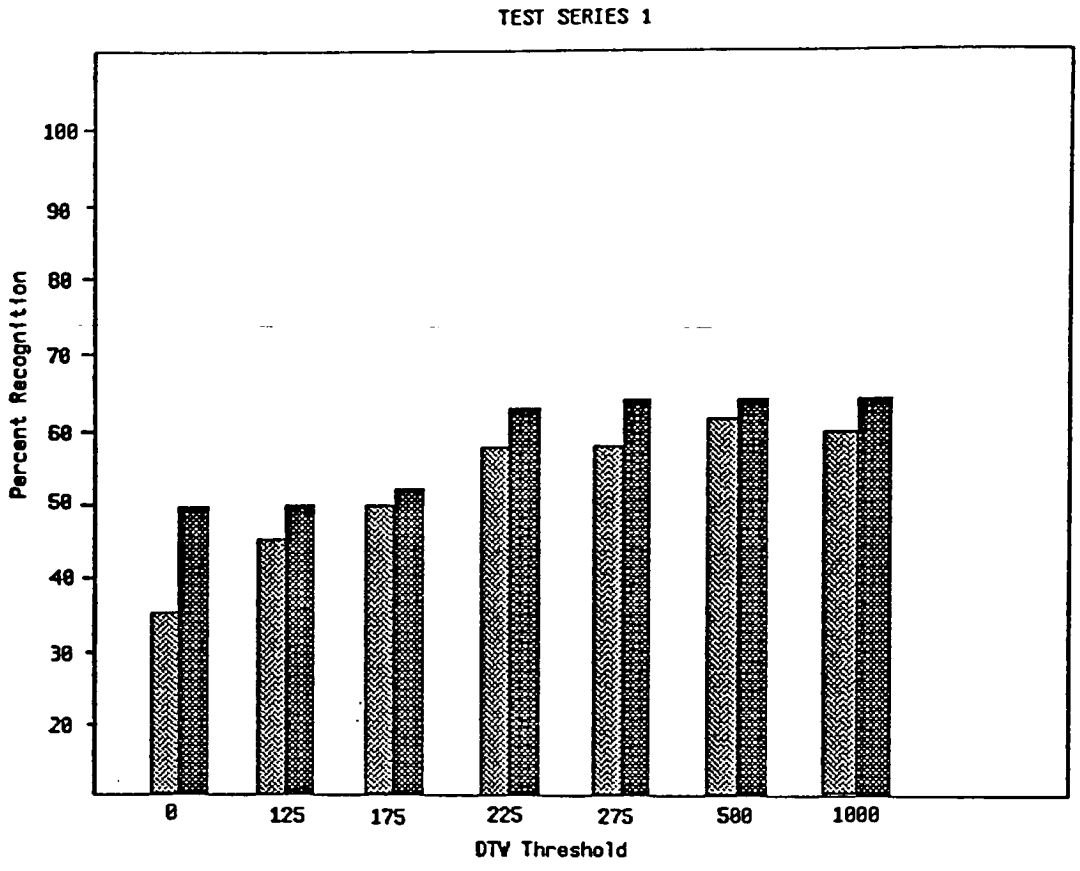


Figure 6: Range of % Total Recognition at Various Given Thresholds Testing on Phrases with Substitution Errors at 10%, Large and Small Lexical Search Criteria, and Candidate Count per Phonetic Index from 5-30

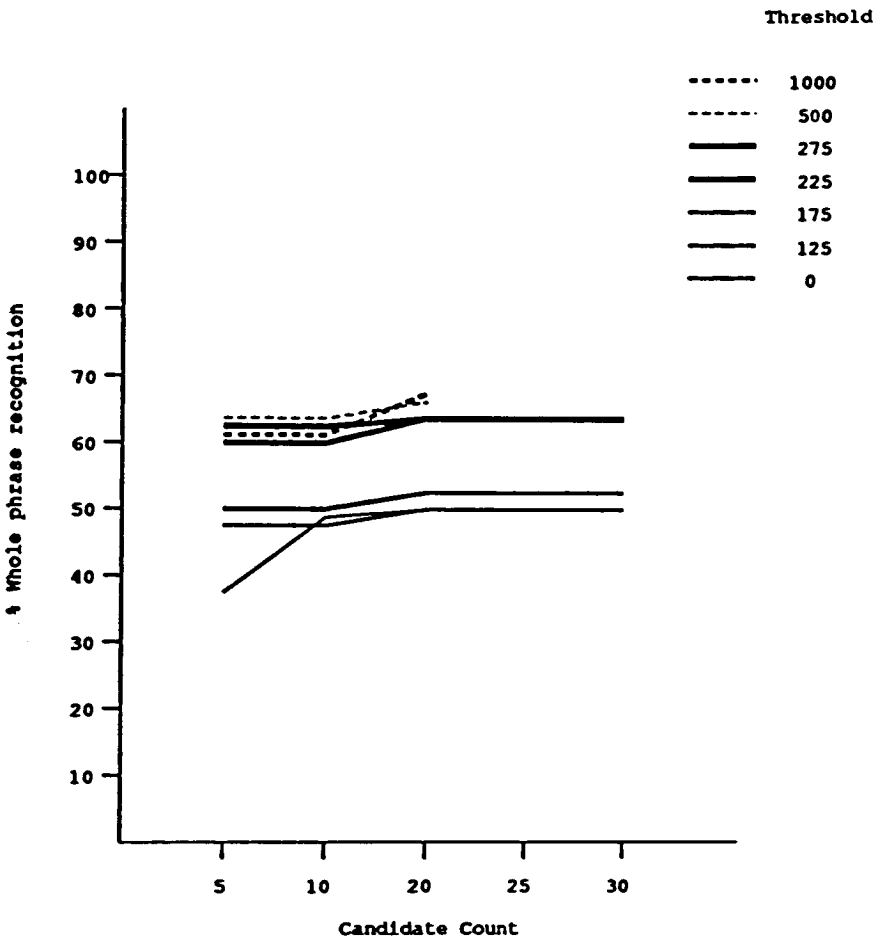


Figure 9: Percent Recognition of Complete Phrase vs. Candidates Accepted per Phonetic Index from Test Series 1: 10% Substitution Errors over Multiple Thresholds Averaged for Large and Small Search Space

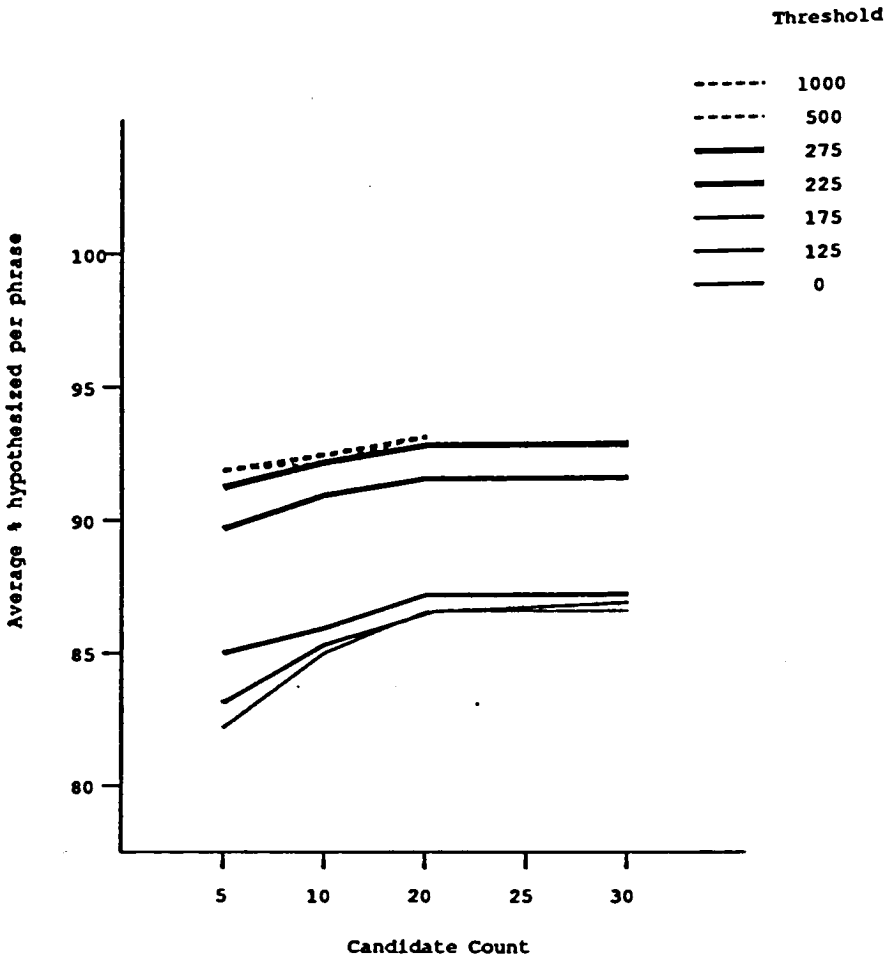


Figure 10: % of Phrase Hypothesized vs Number of Candidates Accepted per Phonetic Index from Test Series 1: 10% Substitution Errors over Multiple Thresholds Averaged for Large and Small Search Space

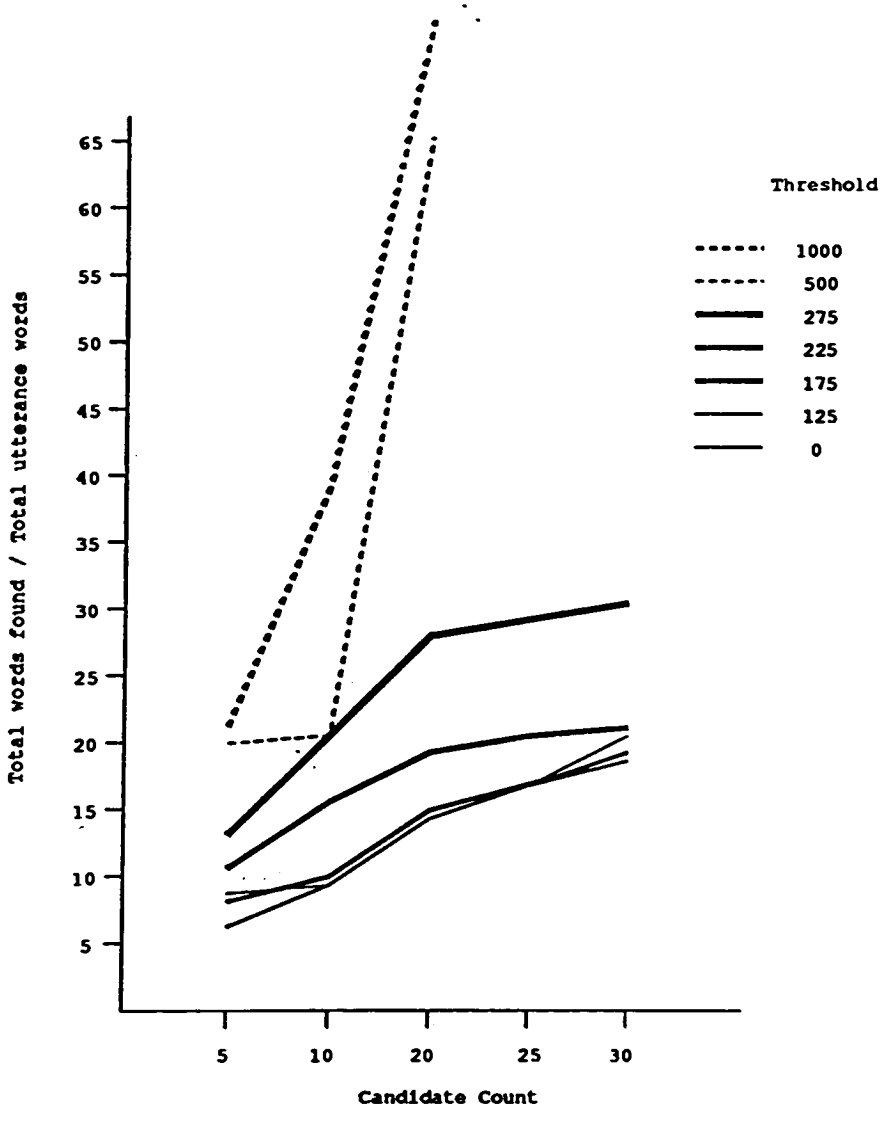


Figure 11: Noise Ratio I vs Number of Word Candidates Accepted per Phonetic Index from Test Series 1: 10% Substitution Errors over Multiple Thresholds Averaged for Large and Small Search Space

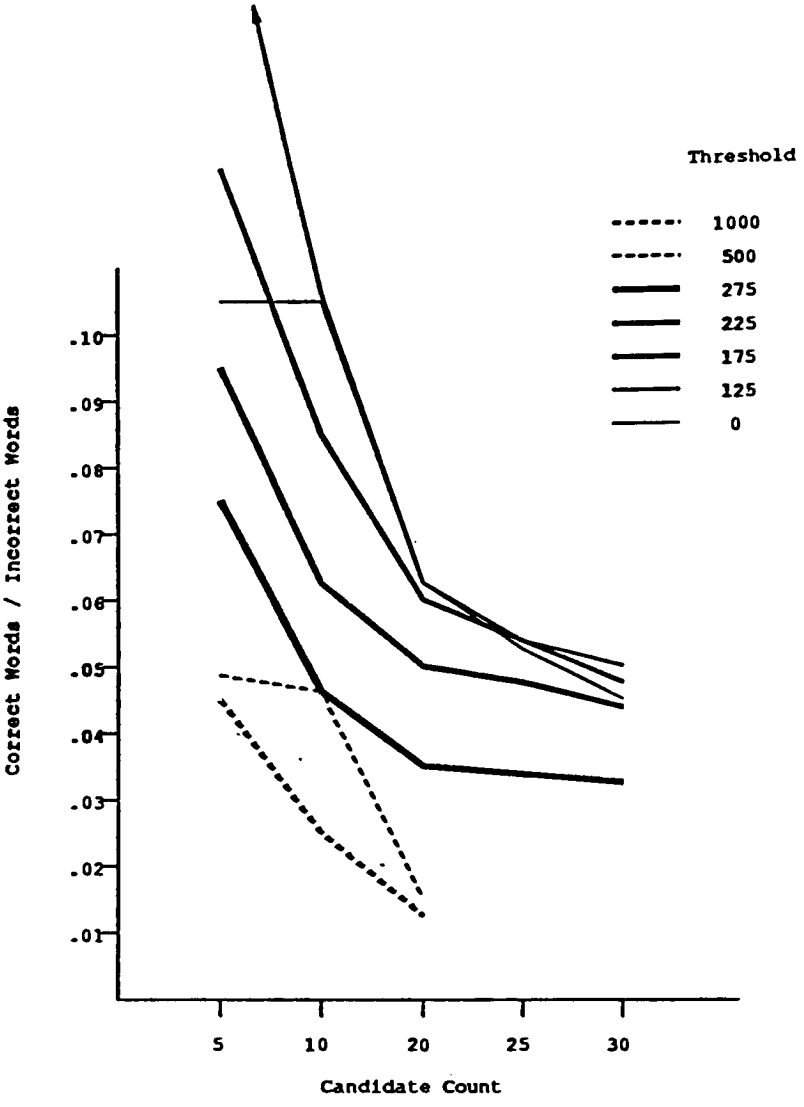


Figure 12: Noise Ratio II vs Number of Word Candidates Accepted per Phonetic Index from Test Series 1: 10% Substitution Errors over Multiple Thresholds Average for Large and Small Search Space

In terms of lexical search space, examination of the Series 1 test results showed that there was no difference in total recognition success until the minimum distance threshold was 225 or above. At these threshold levels, changing from a small to large search space accounted for increases in total phrase recognition of five to six percent. Based on the small increase observed here, the reader is cautioned not to discount the significance of search space size as it relates to total recognition percentages.

Results from the most successful Series 1 test were selected for more thorough examination. Understanding the causes of recognition failure in these results would identify possible changes to the procedure that might improve the total phrase recognition results. The combination of parameter settings that produced the largest percentage of complete matches and the highest average percentage of words hypothesized per phrase was selected for further study. Based on these considerations, the following parameter settings were chosen: (1) a threshold level of 500, (2) 20 candidates allowed per index, and (3) a small lexical search space. With these parameter settings, total recognition rate was 66% and the average percentage hypothesized per phrase was at 93%. Isolating the specific problem areas would be made easier when examining test results with a large average percentage word recognition per phrase. Although large, a noise ratio (total words found vs total utterance words) of 67:1 was accepted in light of the objective to provide a method that would find the entire original phrase from from the errorful phrase.

An immediate observation was that in most cases, words not hypothesized were of short transcription length (under four phonemes). Aware that there are several points within the hypothesis procedure that a potential candidate may be pruned before acceptance, the missed words were submitted individually for parsing and monitored to determine when they were dropped. These locations where pruning may take place include the DTW comparison process, and the two sorting processes based on distance and transcription length.

It was discovered that the DTW comparison procedure was penalizing small words severely and pruning them early on. Early on in empirical testing it was noticed that words with significantly different transcription lengths should be treated differently. As reference pattern length grows, distance generated by a single mismatch in the warping process (when used in comparison to the threshold) has less impact on possible rejection due to averaging. Therefore, the decision to prune shorter words

based on distance must be made earlier than longer words. An adjustment was made to the divisor responsible for averaging distance over reference pattern length (in the case of short transcriptions) and the following test set repeated (threshold at 500 and 1000, twenty candidates per index, small search space).

As is shown in Figure 13 (T1), the percentage for total phrase recognition increased from 66% to 80%. The average percentage hypothesized per phrase increased to approximately 97%. This was accompanied with a thirteen percent increase in the noise ratio of total words found to total utterance words. The results were again examined to determine the reasons that eight phrases remained only partially parsed. It was discovered that if the larger lexical search space had been used, the needed reference patterns would have been supplied for successful DTW of four phrases. Adding these four (now correct) parses to the count of totally correct parses increases the percentage of total phrase recognition to 88% for this test (see Figure 13 - T2). One phrase lost the correctly hypothesized word during the sorting process based on reference length. The three remaining phrases had word-initial errors such that even the larger lexical search did not access the needed reference pattern for comparison. Any further improvement in the process would have to be acheived by enlarging the lexical search to include more reference patterns. Figure 14 shows the percentage total phrase recognition as a function of lexical search space size.

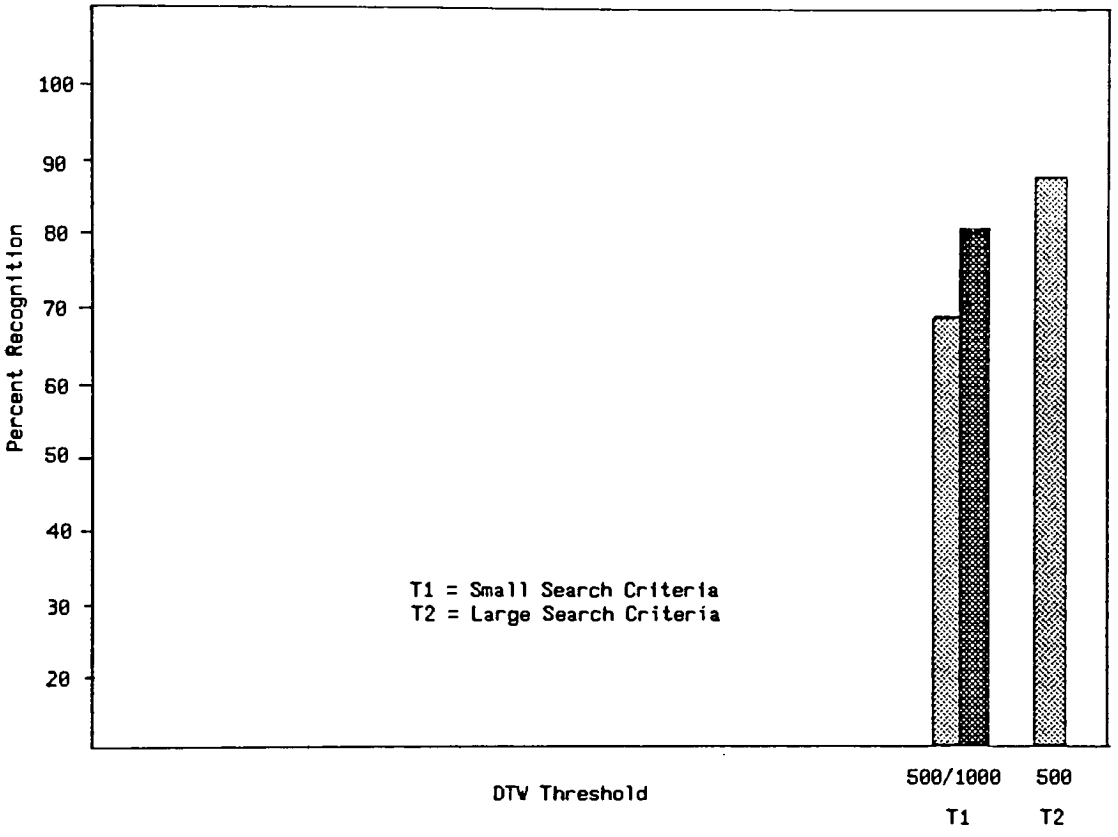


Figure 13: Range of % Total Recognition vs Shown Thresholds Results Reflecting Impact of Change to Threshold Calculation based on Evaluation of Results from Test Series 1. Substitution Errors at 10%, Number of Candidates Allowed per Phonetic Index = 20.

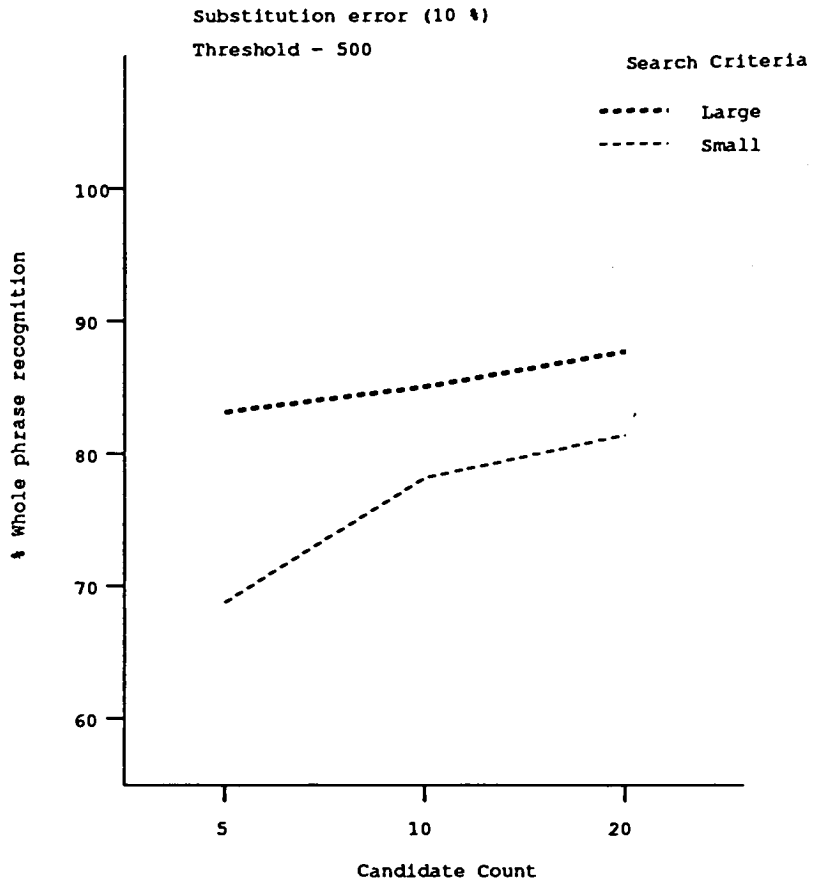


Figure 14: Percent Recognition of Complete Phrase vs. Candidates Accepted per Phonetic Index - Compared over Differential in Search Criteria
Substitution Error Rate - 10% , Threshold - 500

4.2. Test Series 2

A subset of the Series 1 tests was run to confirm the positive effect on recognition of the reduced time warping penalty used in conjunction with short reference patterns. The same distance threshold levels were used, while only a subset of previous values was used for the *candidates per phonetic index* variable (e.g. 5, 20, 30). This series of tests used only the *large* category for the lexical search space variable. It was this value that produced the best results in the previous series of tests.

Figure 15 illustrates the results of these tests. Compared with the Series 1 test results (Figure 8), the percentage of total phrase recognition increased substantially for threshold levels between 125 and 1000. Similar to the Series 1 results, the percentage recognition grew larger as a function of increases in the distance threshold. Increases in recognition stabilized at a maximum level of 88% despite further increases in threshold values. The level zero threshold test did not show as much improvement due to the backtracking and dynamic threshold adjustment that was described previously.

The relationship of total phrase recognition and average percentage hypothesized per phrase to the number of candidates per index is the same as that observed in the Series 1 tests. What differs is the relative level of overall recognition. For the Series 2 tests, this increased between ten and fifteen percent (Figures 16 and 17). The same held true in terms of results representing noise measures of the recognition process (Figures 18 and 19). The noise value representing the effort required to find correct hypotheses increased by a significant amount. In tests with the threshold at 275, ratios of total words found to total words in the utterance were as high as 93:1.

Comparing Series 1 and 2 noise level trends (Figures 11 and 18, respectively) as they related to candidate count, it was discovered that for threshold levels of 275 and below, the Series 2 results had not stabilized as well. This means that increasing the number of candidates accepted at a given index (based on minimum distance and reference length) was not increasing the probability of correct hypothesis. Instead, it was acting to dilute the pool of correct hypotheses which is represented as increased noise levels. This suggests that some factor other than minimum distance comparisons was responsible in finding the correct hypothesis. What was found to play a more significant roll in successful hypothesis is described next.

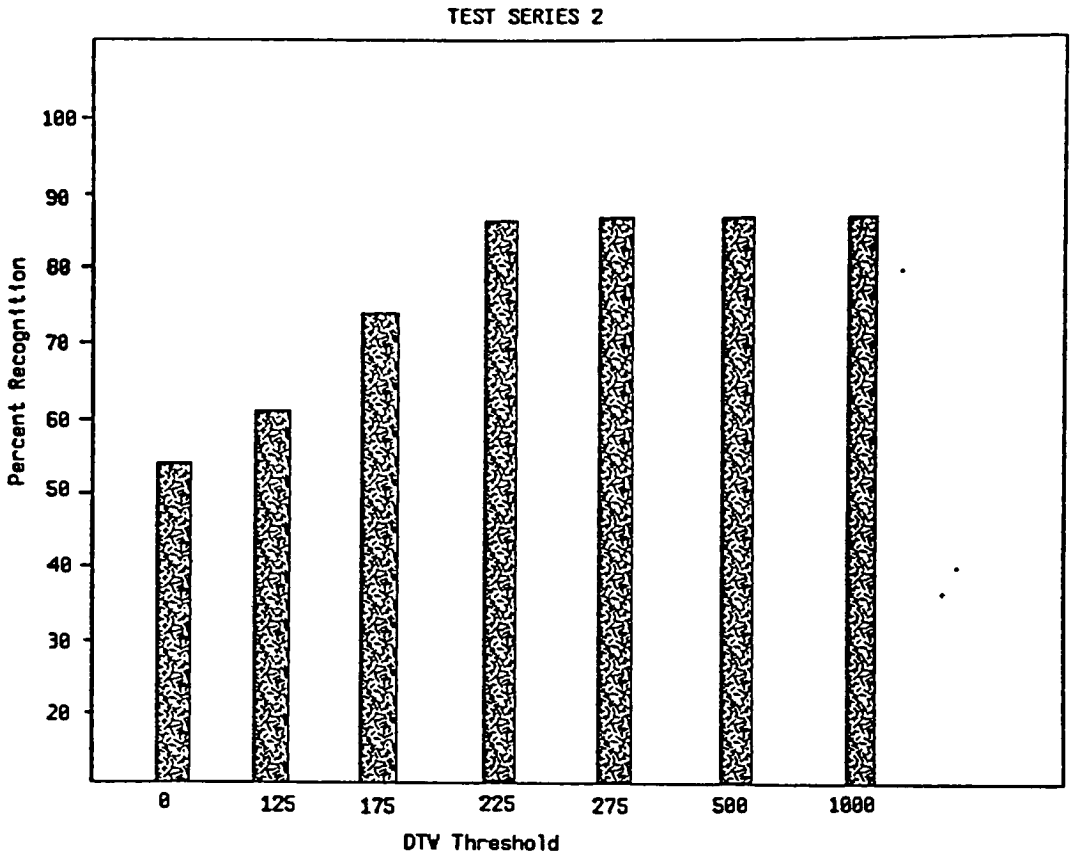


Figure 15: Gains in % Total Recognition vs Threshold as a Result of Changes to a Relaxed Threshold Penalty for Words with Short Transcriptions. Substitution Errors at 10%, Candidates Allowed per Phonetic Index from 5-30, Averaged for Large and Small Search Space

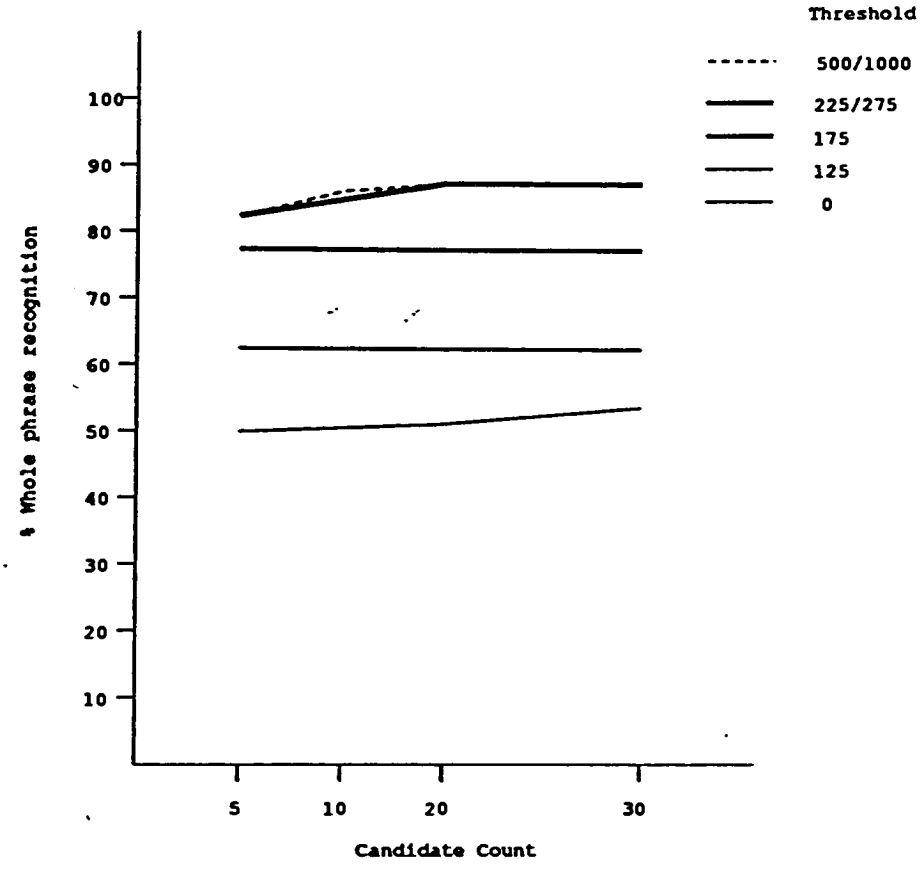


Figure 16: Percent Recognition of Complete Phrase vs. Candidates Accepted per Phonetic Index from Test Series 2: 10% Substitution Errors over Multiple Thresholds Averaged for Large and Small Search Space

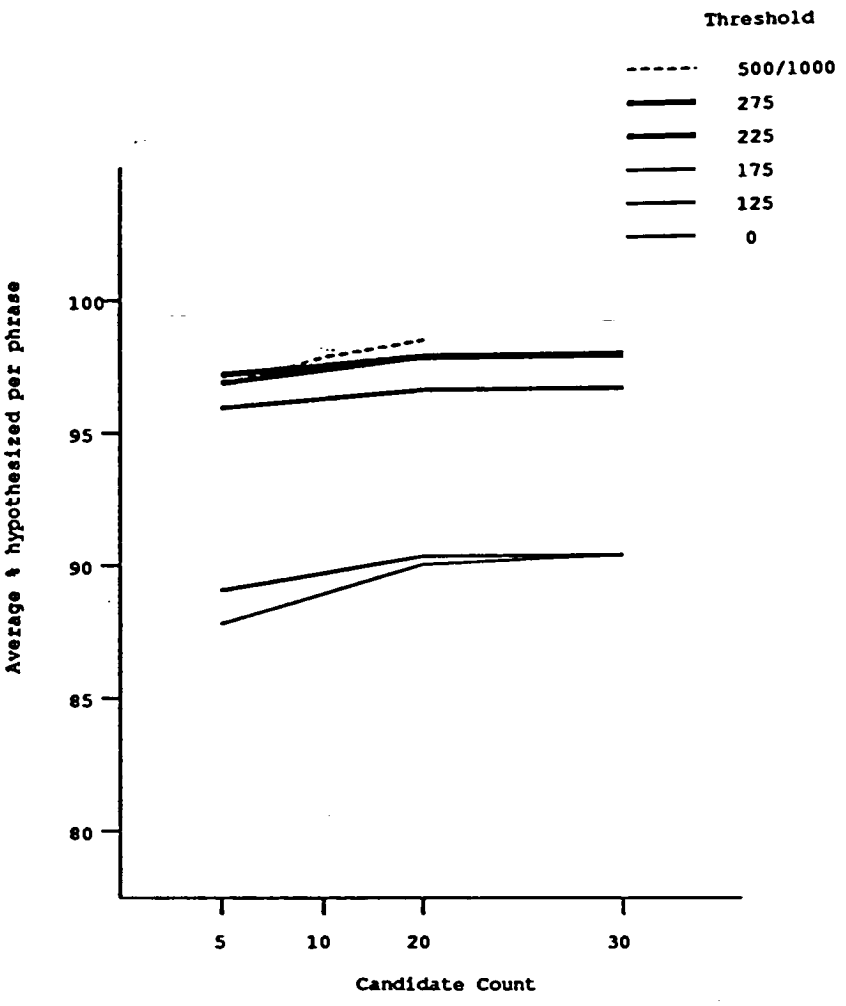


Figure 17: Percent of Phrase Hypothesized vs. Number of Candidates Accepted per Phonetic Index from Test Series 2: 10% Substitution Errors over Multiple Thresholds Averaged for Large and Small Search Space

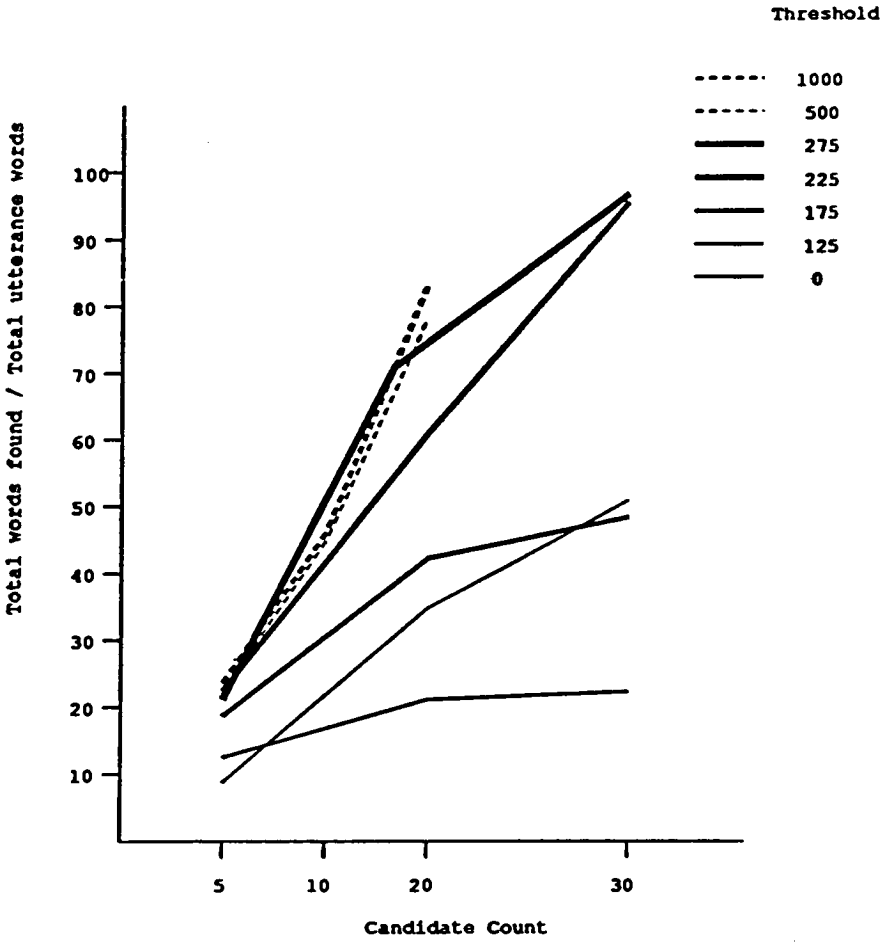


Figure 18: Noise Ratio I vs Number of Word Candidates Accepted per Phonetic Index from Test Series 2: 10% Substitution Errors over Multiple Thresholds Averaged for Large and Small Search Space

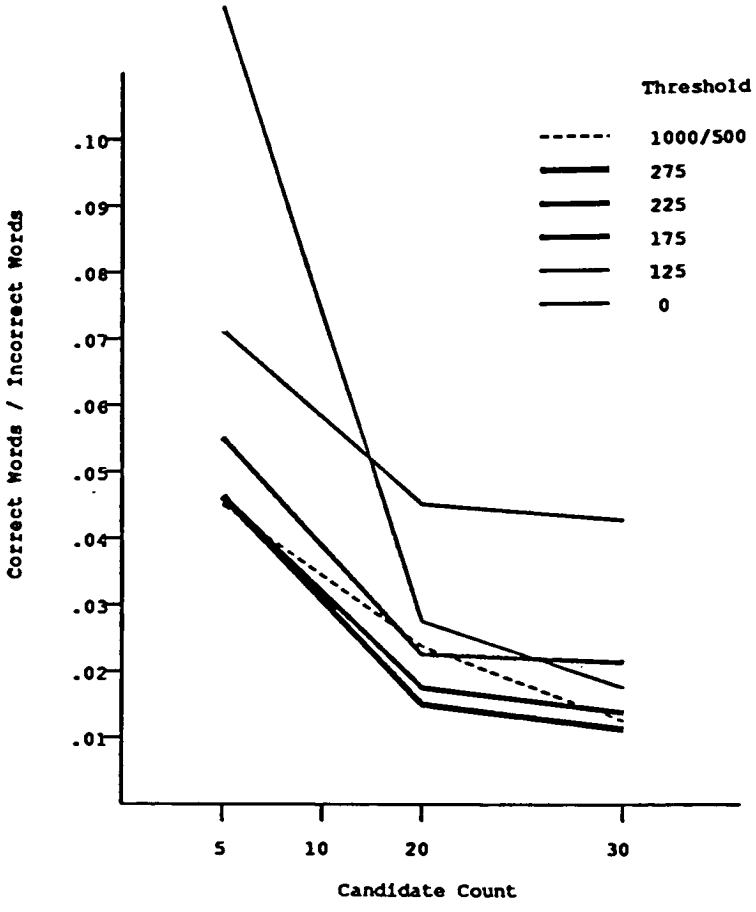


Figure 19: Noise Ratio II vs Number of Word Candidates Accepted per Phonetic Index from Test Series 2: 10% Substitution Errors over Multiple Thresholds Averaged for Large and Small Search Space

Another series of tests was conducted, this time examining the ability of the current algorithm to parse phrases with different percentages of substitution errors. As Figures 20 and 21 illustrate, there was a gradual decrease in the overall level of phrase recognition as the level of errors increased. One contributing factor is that as more errors occur, there is an increase in the probability of exceeding the distance threshold during warping. A more severe problem relates to how reference patterns are selected. Based on the method of reference pattern retrieval from the lexicon, there was a significant chance that the reference pattern would never have an opportunity to be time warped with the unknown. This specific problem accounted for 82 of the failed phrase parses in a test of 148 phrases containing ten percent substitution errors. The remaining phrases not fully parsed reflected another problem. Various reference words had passed the DTW comparison, but a decision was made later to drop the candidate based on rank-order distance and reference pattern length.

Figure 20 also shows the results of a test evaluating the ability of this program to deal with mixed errors at a total rate of 15%. The mix of error types were in the following proportions: insertion and deletion errors each at four percent, and substitution errors at 92%. The ability to completely hypothesize all phrases ranged from 30% to 50%. The two common causes preventing correct hypotheses in the mixed error tests were also observed in the test of the 148 phrase sample containing ten percent substitution errors. Restricted access of reference patterns due to the limitations of the larger lexical search process was one problem. The second problem occurred as hypotheses (passing the warping process) were sorted based on reference length and accumulated warping distance. Out of the 27 phrases not completely parsed in a mixed error test (threshold = 500, candidate count = 5, large lexical search), 62% had found the correct words but then subsequently *trimmed* them from the list of candidates. Many smaller words were hypothesized with zero or low accrued distances, pushing larger words (with comparatively larger distances) out of the candidate list. As was typical with previous tests, enlarging the candidate group would lessen this problem. The problem of not accessing portions of the lexicon containing the necessary reference patterns was exemplified in the remaining forty percent of test phrases not fully parsed.

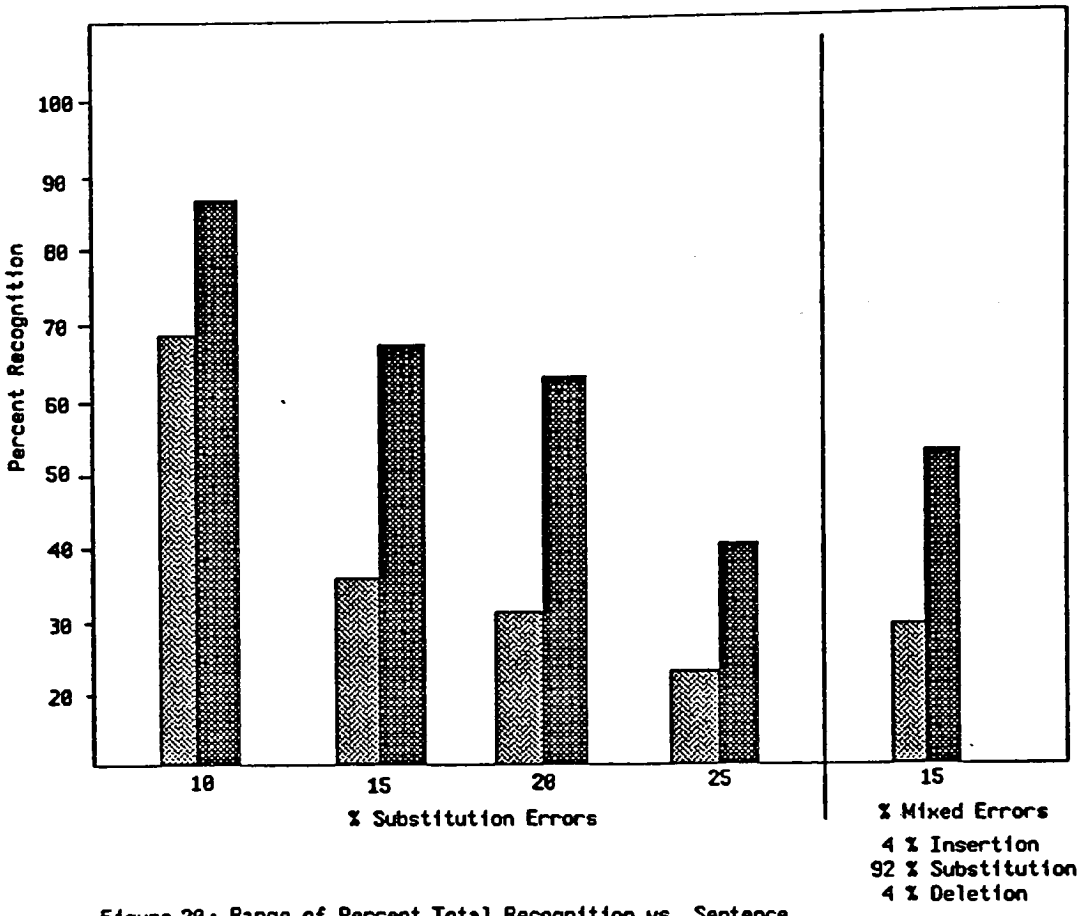


Figure 20: Range of Percent Total Recognition vs. Sentence Error Rate with: Threshold at 500/1000, Candidate Count per Phonetic Index from 5 to 20, and Averaged for Large and Small Search Space

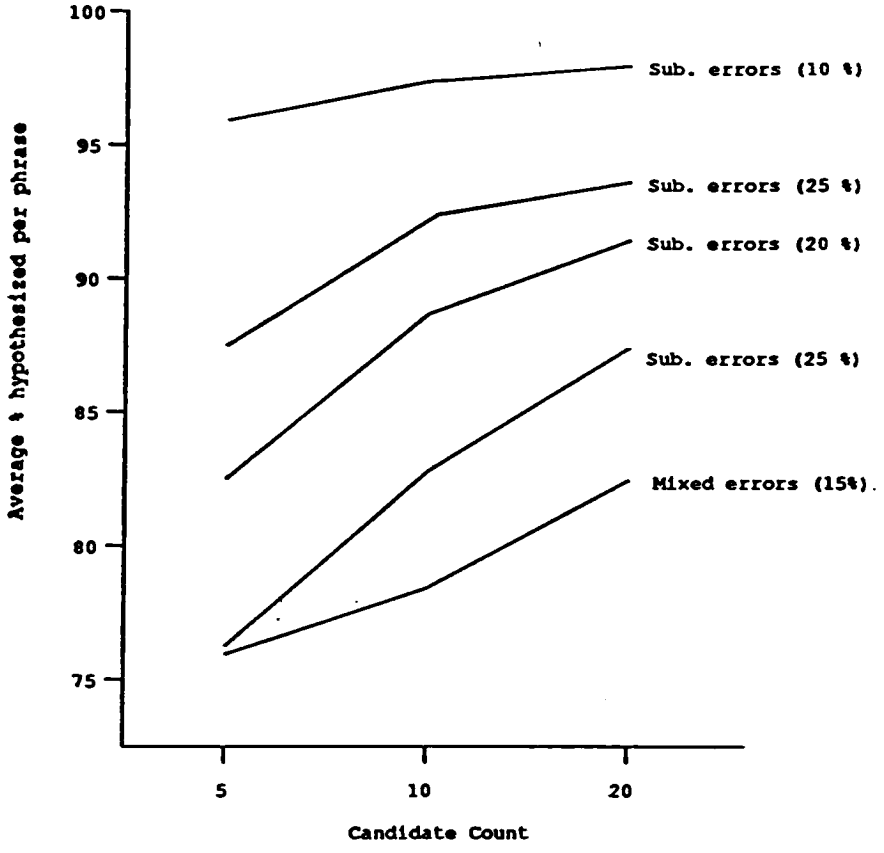


Figure 21: Percent of Phrase Hypothesized vs. Number of Candidates Accepted per Phonetic Index - Results over Range of Error Levels

CHAPTER 5

CONCLUSIONS

From various viewpoints, the dynamic time warping method of this study has proven not to be entirely satisfactory. Examined from the standpoint of real-time speech recognition, the time to process a single phrase was not good. It took approximately three minutes per phrase in the test conditions that provided the greatest percentage of complete phrase recognition. This is several magnitudes greater than human processing times. Some reduction in these processing times might be gained by reducing the code to assembly language, or perhaps embedding it as a utility in firmware. However, as the size of the lexicon increases we would expect the computational resource to grow as well. Assuming the current method of reference pattern selection, the number of DTW comparisons required would increase as each word-initial phoneme group enlarges.

Based on studies of broad phonetic representation by Huttenlocher [HUTT84] and Shipman and Zue [SHIP82], one would expect the similarity of words to increase with increases in lexicon size. Therefore, increased resolution is necessary to determine subtle differences between various references. Without this ability, more candidate words could be hypothesized at a given phonetic index. This presents problems for the syntactic and semantic parsers that would use the word lattice as input.

It was shown that noise levels (measured as the total number of words required to identify a single correct word) experienced during tests demonstrating the best level of hypothesis success, approached 90:1. Large numbers of word candidates would obviously impose severe demands on higher level parsing algorithms. The best performance of word hypothesis occurred at a candidate level of twenty phonemes per index. Assuming that twenty words were hypothesized at each of ten phonetic indices, there are 20^{10} possible phrases in which to find the actual phrase. This places a large burden on the remainder of the recognition procedure. It is hoped that supplying some measure of confidence (distance and reference length) to these upper level processes will be of assistance.

Of the three primary factors tested (threshold, candidate count, lexical selection), variance in distance threshold provided moderate gains in recognition but then quickly leveled out. The remaining two variables were responsible for the majority of missed candidates. Later tests showed that reference pattern selection from the lexicon was too restrictive. In many cases, the reference pattern required for a successful comparison was not included. This in turn is reflective of two more acute problems (1) working without any definitive word boundaries and (2) the possibility of an inaccurate mapping from phonetic distance to confusion probability.

Provision of word boundaries would have an immediate benefit of reducing overall effort. The incremental searching of an unknown phrase gathers many candidates that are not necessary. These candidates then spawn additional sites for continued search. The provision of endpoints would help reduce this by concentrating effort at known intervals.

The selection of reference patterns for DTW comparison assumed that the most potentially confusing words could be associated with the word-initial phoneme of an unknown sequence. It was discovered that in many cases, errors induced at the word-initial phoneme were not within the family of 40 to 60 percent most closely associated phonemes. One might be tempted to incrementally add more phonemes to the list (based on confusion probabilities); however, eventually the list could grow to represent the entire lexicon (i.e., a brute force search). Therefore, some other method is necessary to help guide this procedure. A confidence measure supplied by the lower level phoneme classifiers might be used as a guide for selecting the members of the similarity list. In addition, the establishment of *islands of certainty* via confidence measures might provide points in which to use parallel processing techniques to simultaneously search from several sections of the unknown.

Another procedure that deserves additional consideration is that of pruning the candidate list. Many times in doing so, the correct hypothesis was lost. The sorting procedure as it relates to reference pattern length and distance bears further examination. It was found that often words should be rank-ordered by reference length prior to a sort by accumulated distance. Due to the prevalence of short words and their propensity to have small accumulated distances, many longer candidates were rejected.

Due to the large noise levels required for successful hypothesis at even modest error rates, it is suggested that alternative methods be investigated for the process of word hypothesis in continuous

speech. This could include incorporation of higher level knowledge sources to help prune the searching process. Given a differenct organization or additional information within the lexicon, syntactic and semantic constraints might be used to predict which words are most likely to be present in the unknown.. Another possibility would involve implementing word hypothesis using Hidden Markov Models, comparing the computational resources required. The confusion matrix would provide a way to model the behavior of the system front end.

APPENDICIES

APPENDIX A: Inter-phoneme Distance Matrix used in DTW comparisons. (part 1 of 2)

Input Phoneme	Output Phoneme: er	aa	ae	ah	ao	ax	eh	ih	ix	iy	ow	uh	uw	l	r	y	w
er	0	324	201	181	315	211	170	252	265	570	335	178	285	500	240	670	465
aa	324	0	198	30	101	60	433	704	400	1211	131	205	365	285	604	1300	510
ae	201	198	0	133	396	163	98	282	235	621	420	357	581	600	550	785	740
ah	181	30	133	0	74	30	263	462	262	884	110	102	235	220	360	884	425
ao	315	101	396	74	0	104	501	678	395	1120	60	50	103	165	410	1060	295
ax	211	60	163	30	104	0	293	492	200	910	370	132	265	310	430	1090	480
eh	170	433	98	263	501	293	0	47	360	229	530	336	499	700	340	440	750
ih	252	704	282	462	678	492	47	0	210	75	690	425	543	800	498	270	760
ix	265	400	235	262	395	200	360	210	0	420	370	280	410	610	440	660	630
iy	570	1211	621	884	1120	910	229	75	420	0	1120	765	859	1300	920	450	1000
ow	335	131	420	110	60	370	530	690	370	1120	0	85	123	170	430	1250	270
uh	178	205	357	102	50	132	336	425	280	765	85	0	29	200	330	850	170
uw	285	365	581	235	103	265	499	543	410	859	123	29	0	205	330	859	195
l	500	285	600	220	165	310	700	800	610	1300	170	200	205	0	118	104	145
r	240	604	550	360	410	430	340	498	440	920	430	330	330	118	0	133	110
y	670	1300	785	884	1060	1090	440	270	660	450	1250	850	859	104	133	0	123
w	465	510	740	425	295	480	750	760	630	1000	270	170	195	145	110	123	0
dx	*	*	*	*	*	*	*	*	*	*	*	*	*	345	345	265	280
ng	*	*	*	*	*	*	*	*	*	*	*	*	*	260	195	300	260
b	*	*	*	*	*	*	*	*	*	*	*	*	*	280	205	250	170
t	*	*	*	*	*	*	*	*	*	*	*	*	*	900	820	885	805
d	*	*	*	*	*	*	*	*	*	*	*	*	*	345	345	265	280
k	*	*	*	*	*	*	*	*	*	*	*	*	*	865	775	855	775
g	*	*	*	*	*	*	*	*	*	*	*	*	*	415	405	335	335
m	*	*	*	*	*	*	*	*	*	*	*	*	*	340	275	390	335
n	*	*	*	*	*	*	*	*	*	*	*	*	*	210	185	280	255
p	*	*	*	*	*	*	*	*	*	*	*	*	*	805	710	765	705
hh	*	*	*	*	*	*	*	*	*	*	*	*	*	615	530	565	505
f	*	*	*	*	*	*	*	*	*	*	*	*	*	605	515	585	505
v	*	*	*	*	*	*	*	*	*	*	*	*	*	400	335	355	290
th	*	*	*	*	*	*	*	*	*	*	*	*	*	650	560	625	545
zh	*	*	*	*	*	*	*	*	*	*	*	*	*	590	580	515	505
dh	*	*	*	*	*	*	*	*	*	*	*	*	*	395	345	335	290
s	*	*	*	*	*	*	*	*	*	*	*	*	*	705	625	665	590
z	*	*	*	*	*	*	*	*	*	*	*	*	*	525	490	455	440
sh	*	*	*	*	*	*	*	*	*	*	*	*	*	880	805	835	765
ch	*	*	*	*	*	*	*	*	*	*	*	*	*	885	800	855	775
jh	*	*	*	*	*	*	*	*	*	*	*	*	*	295	255	213	175

APPENDIX A: Inter-phoneme Distance Matrix used in DTW comparisons. (part 2 of 2)

Output Phoneme

Input Phoneme	dx	ng	b	t	d	k	g	m	n	p	hh	f	v	th	zh	dh	s	z	sh	ch	jh
er	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
aa	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
ae	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
ah	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
ao	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
ax	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
eh	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
ih	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
ix	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
iy	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
ow	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
uh	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
uw	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
l	345	260	280	900	345	865	415	340	210	805	615	605	400	650	590	395	705	525	880	885	295
r	345	195	205	820	345	775	405	275	185	710	530	515	335	560	580	345	625	490	805	800	255
y	265	300	250	885	265	855	335	390	280	765	565	585	355	625	515	335	665	455	835	855	213
w	280	260	170	805	280	775	335	335	255	705	505	505	290	545	505	290	590	440	765	775	175
dx	0	530	300	850	30	860	90	595	525	785	550	605	310	610	255	220	585	250	715	800	160
ng	530	0	290	780	530	710	570	95	100	660	540	480	420	550	740	470	650	620	840	790	400
b	300	290	0	635	300	610	315	335	340	540	340	345	135	380	465	180	425	330	605	610	160
t	850	780	635	0	850	130	800	750	870	125	320	310	555	260	825	630	280	660	295	110	740
d	30	530	300	850	0	860	90	595	525	785	550	605	310	610	255	220	585	250	715	800	160
k	860	710	610	130	860	0	820	665	805	80	330	270	555	250	870	645	335	700	410	220	730
g	90	570	315	800	90	820	0	630	580	740	500	575	280	565	180	180	525	165	640	740	180
m	600	95	335	750	595	665	630	0	165	620	540	450	445	525	790	515	645	660	830	770	460
n	530	100	340	875	525	805	580	165	0	755	620	565	470	630	755	505	725	650	910	880	425
p	790	610	540	125	785	80	740	620	755	0	250	200	475	175	790	565	260	620	790	190	660
hh	550	540	340	320	550	330	500	540	620	250	0	140	250	90	560	330	120	380	310	280	420
f	610	480	345	310	605	270	575	450	565	200	140	0	300	80	655	395	225	485	405	320	470
v	310	420	135	555	310	555	280	445	470	475	250	300	0	305	390	100	310	230	480	520	180
th	610	550	380	260	610	250	565	525	630	175	90	80	305	0	630	390	155	460	325	250	490
zh	260	740	460	825	255	870	180	790	755	790	560	655	390	630	0	295	545	175	610	750	350
dh	220	470	180	630	220	645	180	515	505	565	330	395	100	390	295	0	365	150	515	580	130
s	590	650	425	280	585	335	525	645	725	260	120	225	310	155	545	365	0	385	190	220	490
z	250	620	330	660	250	700	165	660	650	620	380	485	230	460	175	150	385	0	480	590	250
sh	720	840	605	295	715	410	640	830	910	355	310	405	480	325	610	515	190	480	0	220	640
ch	800	790	610	110	800	220	740	770	880	190	280	320	520	250	750	580	220	590	210	0	690
jh	160	400	155	740	160	730	175	460	430	660	420	470	180	490	345	125	490	250	640	690	0

APPENDIX B: Inter-phoneme Confusion Matrix derived from Distance Matrix, and used in the creation of test phrases. (part 1 of 2)

Input Phoneme	Output Phoneme: er	aa	ae	ah	ao	ax	eh	ih	ix	iy	ow	uh	uw	l	r	y	w
er	-	5.2	8.5	9.4	5.4	8.1	10.0	6.7	6.4	3.0	5.1	9.6	6.0	3.4	7.1	2.5	3.7
aa	3.1	-	5.1	33.9	10.1	17.0	2.4	1.4	2.5	0.8	7.8	5.0	2.8	3.6	1.7	0.8	2.0
ae	8.5	8.6	-	12.8	4.3	10.4	17.3	6.0	7.2	2.7	4.0	4.8	2.9	2.8	3.1	2.2	2.3
ah	4.0	24.1	5.4	-	9.8	24.1	2.8	1.6	2.8	0.8	6.6	7.1	3.1	3.3	2.0	0.8	1.7
ao	3.0	9.4	2.4	12.9	-	9.2	1.9	1.4	2.4	0.9	15.9	19.1	9.3	5.8	2.3	0.9	3.2
ax	4.5	15.9	5.9	31.9	9.2	-	3.3	1.9	4.8	1.1	2.6	7.2	3.6	3.1	2.2	0.9	2.0
eh	8.3	3.3	14.4	5.4	2.8	4.8	-	30.0	3.9	6.2	2.7	4.2	2.8	2.0	4.2	3.2	1.9
ih	5.8	2.1	5.2	3.2	2.2	3.0	31.3	-	7.0	19.6	2.1	3.5	2.7	1.8	3.0	5.5	1.9
ix	7.9	5.3	9.0	8.0	5.3	10.5	5.8	10.0	-	5.0	5.7	7.5	5.1	3.4	4.8	3.2	3.3
iy	4.9	2.3	4.5	3.2	2.5	3.1	12.2	37.2	6.6	-	2.5	3.6	3.2	2.1	3.0	6.2	2.8
ow	3.7	9.4	2.9	11.2	20.6	3.3	2.3	1.8	3.3	1.1	-	14.5	10.0	7.3	2.9	1.0	4.6
uh	4.6	4.0	2.3	8.0	16.4	6.2	2.4	1.9	2.9	1.1	9.6	-	28.2	4.1	2.5	1.0	4.8
uw	3.9	3.0	1.9	4.7	10.8	4.2	2.2	2.0	2.7	1.3	9.0	38.3	-	5.4	3.[illegible]	1.3	5.7
l	1.8	3.1	1.5	4.0	5.3	2.8	1.3	1.1	1.4	0.7	5.2	4.4	4.3	-	7.4	8.4	6.0
r	3.7	1.5	1.6	2.5	2.2	2.1	2.6	1.8	2.0	1.0	2.1	2.7	2.7	7.6	-	6.7	8.2
y	1.6	0.8	1.3	1.2	1.0	1.0	2.4	3.9	1.6	2.3	0.8	1.2	1.2	10.0	7.8	-	8.5
w	1.9	1.7	1.2	2.0	2.9	1.8	1.2	1.1	1.4	0.9	3.2	5.1	4.5	6.0	7.9	7.1	-
dx	-	-	-	-	-	-	-	-	-	-	-	-	-	2.9	2.9	3.7	3.5
ng	-	-	-	-	-	-	-	-	-	-	-	-	-	5.5	7.4	4.8	5.5
b	-	-	-	-	-	-	-	-	-	-	-	-	-	4.4	6.0	4.9	7.3
t	-	-	-	-	-	-	-	-	-	-	-	-	-	1.8	1.9	1.8	2.0
d	-	-	-	-	-	-	-	-	-	-	-	-	-	2.9	2.9	3.7	3.5
k	-	-	-	-	-	-	-	-	-	-	-	-	-	1.9	2.1	1.9	2.1
g	-	-	-	-	-	-	-	-	-	-	-	-	-	2.9	3.0	3.7	3.7
m	-	-	-	-	-	-	-	-	-	-	-	-	-	4.9	6.0	4.3	5.0
n	-	-	-	-	-	-	-	-	-	-	-	-	-	7.5	8.5	5.6	6.1
p	-	-	-	-	-	-	-	-	-	-	-	-	-	1.8	2.0	1.9	2.0
hh	-	-	-	-	-	-	-	-	-	-	-	-	-	2.1	2.4	2.3	2.5
f	-	-	-	-	-	-	-	-	-	-	-	-	-	2.2	2.6	2.3	2.7
v	-	-	-	-	-	-	-	-	-	-	-	-	-	3.1	3.6	3.4	4.2
th	-	-	-	-	-	-	-	-	-	-	-	-	-	1.9	2.2	1.9	2.2
zh	-	-	-	-	-	-	-	-	-	-	-	-	-	3.1	3.2	3.6	3.6
dh	-	-	-	-	-	-	-	-	-	-	-	-	-	3.0	3.4	3.5	4.0
s	-	-	-	-	-	-	-	-	-	-	-	-	-	2.0	2.3	2.1	2.4
z	-	-	-	-	-	-	-	-	-	-	-	-	-	2.8	3.0	3.2	3.3
sh	-	-	-	-	-	-	-	-	-	-	-	-	-	2.2	2.4	2.4	2.6
ch	-	-	-	-	-	-	-	-	-	-	-	-	-	1.9	2.0	1.9	2.1
jh	-	-	-	-	-	-	-	-	-	-	-	-	-	4.0	4.6	5.5	6.7

APPENDIX B: Inter-phoneme Confusion Matrix derived from Distance Matrix, and used in the creation of test phrases. (part 2 of 2)

Input Phoneme	Output Phoneme: dx	ng	b	t	d	k	g	m	n	p	hh	f	v	th	zh	dh	s	z	sh	ch	jh
er	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
aa	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ae	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ah	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ao	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ax	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eh	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ih	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ix	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
iy	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ow	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
uh	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
uw	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
l	2.5	3.4	3.1	1.0	2.5	1.0	2.1	2.6	4.2	1.1	1.4	1.4	2.2	1.3	1.5	2.2	1.2	1.7	1.0	1.0	3.0
r	2.6	4.6	4.4	1.1	2.6	1.2	2.2	3.3	4.8	1.3	1.7	1.7	2.7	1.6	1.5	2.6	1.4	1.8	1.1	1.1	3.5
y	3.9	3.5	4.2	1.2	3.9	1.2	3.1	2.7	3.7	1.4	1.8	1.8	2.9	1.7	2.0	3.1	1.6	2.3	1.2	1.2	4.9
w	3.1	3.3	5.1	1.1	3.1	1.1	2.6	2.6	3.4	1.2	1.7	1.7	3.0	1.6	1.7	3.0	1.5	2.0	1.1	1.1	5.0
dx	-	1.9	3.3	1.2	32.9	1.1	11.0	1.7	1.9	1.3	1.8	1.6	3.2	1.6	3.9	4.5	1.7	3.9	1.4	1.2	6.2
ng	2.7	-	5.0	1.8	2.7	2.0	2.5	15.1	14.4	2.2	2.7	3.0	3.4	2.6	1.9	3.1	2.2	2.3	1.7	1.8	3.6
b	4.1	4.3	-	1.9	4.1	2.0	3.9	3.7	3.6	2.3	3.6	3.6	9.1	3.2	2.7	6.8	2.9	3.7	2.0	2.0	7.7
t	1.9	2.0	2.5	-	1.9	12.2	2.0	2.1	1.8	12.7	5.0	5.1	2.9	6.1	1.9	2.5	5.7	2.4	5.4	14.4	2.1
d	32.9	1.9	3.3	1.2	-	1.1	11.0	1.7	1.9	1.3	1.8	1.6	3.2	1.6	3.9	4.5	1.7	3.9	1.4	1.2	6.2
k	1.9	2.3	2.6	12.3	1.9	-	2.0	2.4	2.0	20.0	4.9	5.9	2.9	6.4	1.8	2.5	4.8	2.3	3.9	7.3	2.2
g	13.6	2.1	3.9	1.5	13.6	1.5	-	1.9	2.1	1.7	2.4	2.1	4.4	2.2	6.8	6.8	2.3	7.4	1.9	1.7	6.8
m	2.8	17.5	5.0	2.2	2.8	2.5	2.6	-	10.1	2.7	3.1	3.7	3.7	3.2	2.1	3.2	2.6	2.5	2.0	2.2	3.6
n	3.0	15.7	4.6	1.8	3.0	1.9	2.7	9.5	-	2.1	2.5	2.8	3.3	2.5	2.1	3.1	2.2	2.4	1.7	1.8	3.7
p	1.8	2.4	2.7	11.5	1.8	18.0	1.9	2.3	1.9	-	5.8	7.2	3.0	8.2	1.8	2.5	5.5	2.3	1.8	7.6	2.2
hh	2.3	2.4	3.7	4.0	2.3	3.9	2.5	2.4	2.1	5.1	-	9.1	5.1	14.1	2.3	3.9	10.6	3.4	4.1	4.5	3.0
f	2.2	2.8	3.9	4.3	2.2	5.0	2.3	3.0	2.4	6.7	9.6	-	4.5	16.8	2.0	3.4	6.0	2.8	3.3	4.2	2.9
v	3.9	2.9	9.0	2.2	3.9	2.2	4.4	2.7	2.6	2.6	4.9	4.1	-	4.0	3.1	12.2	3.9	5.3	2.5	2.3	6.8
th	2.0	2.2	3.2	4.7	2.0	4.9	2.2	2.3	1.9	7.0	13.5	15.2	4.0	-	1.9	3.1	7.9	2.6	3.7	4.9	2.5
zh	7.0	2.5	4.0	2.2	7.2	2.1	10.2	2.3	2.4	2.3	3.3	2.8	4.7	2.9	-	6.2	3.4	10.5	3.0	2.4	5.2
dh	5.3	2.5	6.5	1.9	5.3	1.8	6.5	2.3	2.3	2.1	3.6	3.0	11.7	3.0	4.0	-	3.2	7.8	2.3	2.0	9.0
s	2.4	2.2	3.4	5.1	2.4	4.3	2.7	2.2	2.0	5.5	11.9	6.3	4.6	9.2	2.6	3.9	-	3.7	7.5	6.5	2.9
z	5.8	2.3	4.4	2.2	5.8	2.1	8.8	2.2	2.2	2.3	3.8	3.0	6.3	3.2	8.3	9.7	3.8	-	3.0	2.5	5.8
sh	2.7	2.3	3.2	6.7	2.7	4.8	3.1	2.4	2.2	5.5	6.3	4.8	4.1	6.0	3.2	3.8	10.3	4.1	-	8.9	3.1
ch	2.0	2.1	2.7	14.9	2.0	7.4	2.2	2.1	1.9	8.6	5.8	5.1	3.1	6.6	2.2	2.8	7.4	2.8	7.8	-	2.4
jh	7.3	2.9	7.5	1.6	7.3	1.6	6.7	2.5	2.7	1.8	2.8	2.5	6.5	2.4	3.4	9.3	2.4	4.7	1.8	1.7	-

APPENDIX C: Phonetic Symbols DECtalk[1] vs Carnegie-Mellon University

VOWELS

DEC	CMU	example
rr	er	bird
yu	ux	beauty
aa	†	cot
ae	†	bat
ah	†	butt
ao	†	bought
aw	†	bough
ax	†	the
ay	†	bite
eh	†	bet
ey	†	bait
ih	†	bit
ix	†	roses
iy	†	beat
oy	†	boy
ow	†	boat
uh	†	book
uw	†	boot
ir	ih r	beet
ar	aa r	bar
or,ur	ao r	poor
er	eh r	bare

LIQUIDS

DEC	CMU	example
l	†	led
r	†	red

GLIDES

DEC	CMU	example
y	†	yet
w	†	wet

FLAPS

DEC	CMU	example
dx	†	rider

STOPS

DEC	CMU	example
nx	ng	sing
b	†	bob
t	†	tot
d	†	dad
k	†	kick
g	†	gag
m	†	mom
n	†	non
p	†	pop

FRICATIVES

DEC	CMU	example
hx	hh	hay
f	†	fief
v	†	verv
th	†	thief
zh	†	measure
dh	†	they
s	†	sis
z	†	zoo
sh	†	shoe

AFFRICATES

DEC	CMU	example
ch	†	church
jh	†	judge

SYLLABIC RESONANTS

DEC	CMU	example
el	†	bottle
em	†	ransom
en	†	button

† indicates equivalent symbol used

[1]DECtalk is a trademark of Digital Equipment Corporation

Appendix D: Test Phrases Created with 10% Substitution Errors

Template for Each Entry:

Phrase Transcription with Errors
Phrase Transcription with No Errors
English Representation of Phrase

(T AX IH M AE N D L OW P EH IH SH AX N AX V R AA N D IX V UW)
(T AA IH M AE N D L OW K EH IH SH AX N AX V R AA N D IX V UW)
(TIME AND LOCATION OF RENDEZVOUS)

(AH P D EH IH T DH AX AE L AA IH D V T EH IH P AX S)
(AH P D EH IH T DH AX AE L AA IH D S T EH IH T AX S)
(UPDATE THE ALLIED STATUS)

(W EH R R R DH AX AH DH ER P L EH IH N Z)
(W EH R AA R DH AX AH DH ER P L EH IH N Z)
(WHERE ARE THE OTHER PLANES)

(R IY K W EH S T M AA IH AA P SH AX N Z AE N D R AA N D IX V UW D EH IH T OW)
(R IY K W EH S T M AA IH AA P SH AX N Z AE N D R AA N D IX V UW D EH IH T AX)
(REQUEST MY OPTIONS AND RENDEZVOUS DATA)

(D IX S P L EH IX HH OW R IX Z AA M T L S IH CH R EH IH SH N AX V S F R AA IH K ER Z)
(D IX S P L EH IH HH OW R IX Z AA N T L S IH CH UW EH IH SH N AX V S T R AA IH K ER Z)
(DISPLAY HORIZONTAL SITUATION OF STRIKERS)

(P AO IH N T N AH M B ER AE L AA IH D S T EH IH CH AX S)
(P AO IH N T N AH M B ER AE L AA IH D S T EH IH T AX S)
(POINT NUMBER ALLIED STATUS)

(S T EH IH T D EH IH T AX AE NG D SH OW M IY AX T AE K F L AA IH T)
(S T EH IH T D EH IH T AX AE N D SH OW M IY AX T AE K F L AA IH T)
(STATE DATA AND SHOW ME ATTACK FLIGHT)

(S T EH IH T AX S AX V SH T DX AA IH K F OW AA IH T)
(S T EH IH T AX S AX V S T R AA IH K F L AA IH T)
(STATUS OF STRIKE FLIGHT)

(K AE N AA IH K IH L HH IH Z UW R K AE DX AA IH AX V AO IH D HH IH M)
(K AE N AA IH K IH L HH IH M OW R K AE N AA IH AX V AO IH D HH IH M)
(CAN I KILL HIM OR CAN I AVOID HIM)

(D IX CH K R AA IH B TH R EH T AE N D D IX S P L EH IH TH R EH T R UW IH D IY AX S)
(D IX S K R AA IH B TH R EH T AE N D D IX S P L EH IH TH R EH T R EH IH D IY AX S)
(DESCRIBE THREAT AND DISPLAY THREAT RADIUS)

(G IH NG M IY M OW R IX N F ER K EH IH SH IX N AA N DH AX TH R EH T)
(G IH V M IY M OW R IX N F ER M EH IH SH IX N AA N DH AX TH R EH T)
(GIVE ME MORE INFORMATION ON THE THREAT)

(IH Z IH T IH N AE N AE K T IX V M OW DX)
(IH Z IH T IH N AE N AE K T IX V M OW D)
(IS IT IN AN ACTIVE MODE)

(P R EH S AX N T DH AX TH R EH T D EH IH T AX)
(P R EH S AX N T DH AX TH R EH T D EH IH T AX)
(PRESENT THE THREAT DATA)

(N OW T AX F AA IH W IH NG G M AX N T UH S T AH R T DH AX IH N T AA S EH P T)
(N OW T AX F AA IH W IH NG G M AX N T UW S T AA R T DH AX IH N T ER S EH P T)
(NOTIFY WINGMAN TO START THE INTERCEPT)

(R EH N IH N R EH EH N JH L AX K AE N D IX N F OW R M M IY)
(W EH N IH N R EH IH N JH L AA K AE N D IX N F OW R M M IY)
(WHEN IN RANGE LOCK AND INFORM ME)

(AA AE M T UW M IH S L Z N IH V M IH IH NG R EH IH N JH AA N B OW TH)
(AA R M T UW M IH S L Z G IH V M IY IH N R EH IH N JH AA N B OW TH)
(ARM TWO MISSILES GIVE ME IN RANGE ON BOTH)

(AA IH T EH IH K DH AX N UW R AA UH T AE ZH ZH F EH L DH AX F L AA IH T)
(AA IH T EH IH K DH AX N UW R AA UH T AE N D T EH L DH AX F L AA IH T)
(I TAKE THE NEW ROUTE AND TELL THE FLIGHT)

(TH EH L DH AX R EH S T AX V G AX F R AA IH TH)
(T EH L DH AX R EH S T AX V DH AX F L AA IH T)
(TELL THE REST OF THE FLIGHT)

(D IX S P L EH IY S AX L EH K T IX M AX T AE T JH IY AA M AX T R IY)
(D IX S P L EH IH S AX L EH K T IX D AX T AE K JH IY AA M AX T R IY)
(DISPLAY SELECTED ATTACK GEOMETRY)

(L AA K AA N T AA R G UW T AA N DH AX N ER Z TH ER T IY F AA IY V M AA IY L Z)
(L AA K AA N T AA R G IX T AA N DH AX N OW Z TH ER T IY F AA IH V M AA IH L Z)
(LOCK ON TARGET ON THE NOSE THIRTY FIVE MILES)

(N AE V M AE P AX K S P AE N DX AA N EH L AA R EH HH TH R EH T)
(N AE V M AE P AX K S P AE N D AA N EH L AA R EH S TH R EH T)
(NAV MAP EXPAND ON LRS THREAT)

(W AX T K AA EH N D AX V M IH S L D D AX AA IH HH AE V)
(W AX T K AA IH N D AX V M IH S L Z D UW AA IH HH AE V)
(WHAT KIND OF MISSILES DO I HAVE)

(CH AE F F L EH R Z S ER L F OW T UW S EH K AX N D Z)
(CH AE F F L EH R Z S AE L V OW T UW S EH K AX N D Z)
(CHAFF FLARES SALVO TWO SECONDS)

(G IH F M IY JH AH M IX NG AE N D CH AE F)
(G IH V M IY JH AE M IX NG AE N D CH AE F)
(GIVE ME JAMMING AND CHAFF)

(R EH IH N JH EH M D B EH IH R IX NG AX B S CH R AA IH F ER Z)
(R EH IH N JH AE N D B EH IH R IX NG AX V S T R AA IH K ER Z)
(RANGE AND BEARING OF STRIKERS)

(R IY K W EH S T F AA IH T ER P AX S IH SH AX N)
(R IY K W EH S T F AA IH T ER P AX S IH SH AX N)
(REQUEST FIGHTER POSITION)

(SH OW M EH EH N IY HH AA IH D EH IH N JH ER TH R AA T S AE N D EH R T UW EH R TH R EH T S)
(SH OW M IY EH N IY HH AA IH D EH IH N JH ER TH R EH T S AE N D EH R T UW EH R TH R EH T S)
(SHOW ME ANY HIGH DANGER THREATS AND AIR TO AIR THREATS)

(R EH OW D AH R EH JH R ER T D AE K W AA IH L S K AH N T AA R G IX T HH EH L AX K AA P T ER)
(R EH IH D AA R EH N T ER T R AE K W AA IH L S K AE N T AA R G IX T HH EH L AX K AA P T ER)
(RADAR ENTER TRACK WHILE SCAN TARGET HELICOPTER)

(W AX T K AA IH N DX AX V B IH S L Z D UW AA IH HH AE V)
(W AX T K AA IH N D AX V M IH S L Z D UW AA IH HH AE V)
(WHAT KIND OF MISSILES DO I HAVE)

(D IX S P T EH IH R EH P AH N Z P R AE M AX T ER Z)
(D IX S P L EH IH W EH P AX N Z P ER AE M AX T ER Z)
(DISPLAY WEAPONS PARAMETERS)

(G IH V M IY N IH S L T AA R G IX T D IX S IX G N EH IH SH AX N AA N W IY N T W T EH IH K DH AX SH AA T)
(G IH V M IY M IH S L T AA R G IX T D IX S IX G N EH IH SH AX N AA N W EH N T UW T EH IH K DH AX SH AA T)
(GIVE ME MISSILE TARGET DESIGNATION ON WHEN TO TAKE THE SHOT)

(AO L T ER K OW R S T UW R IY R AA UH T N OW T AX F AA IH B IH NG G M AX N AE N D P AE K IX JH)
(AO L T ER K OW R S T UW R IY R AA UH T N OW T AX F AA IH W IH NG G M AX N AE N D P AE K IX JH)
(ALTER COURSE TO REROUTE NOTIFY WINGMAN AND PACKAGE)

(AA IH T EH IH K DH AX N UW R AH UH T AE N D T EH L DH AX F L AO IH T)
(AA IH T EH IH K DH AX N UW R AA UH T AE N D T EH L DH AX F L AA IH T)
(I TAKE THE NEW ROUTE AND TELL THE FLIGHT)

(F UW K OW ZH S S AX L EH K T IX D D EH IH T R L IH NG T)
(N UW K OW R S S AX L EH K T IX D D EH IH T AX L IH NG K)
(NEW COURSE SELECTED DATA LINK)

(P AE S R IY R AA UH T IH N F OW T UW F OW R M EH IH SH AX N)
(P AE S R IY R AA UH T IH N F OW T UW F OW R M EH IH SH AX N)
(PASS REROUTE INFO TO FORMATION)

(S EH N D HH OW R IX Z AO N T L S IH V UW EH IH SH N T UW AH DH ER F L AA IH T M EH M B ER JH)
(S EH N D HH OW R IX Z AA N T L S IH CH UW EH IH SH N T UW AH DH ER F L AA IH T M EH M B ER Z)
(SEND HORIZONTAL SITUATION TO OTHER FLIGHT MEMBERS)

(S IH N D N UW D EH IH T AX P UW W IH NG G M AX N F L AA IH T)
(S EH N D N UW D EH IH T AX T UW W IH NG G M AX N F L AA IH T)
(SEND NEW DATA TO WINGMAN FLIGHT)

(EH R T AE IH R S EH L EH S T TH P Y R OW AA R M)
(EH R T UW EH R S EH L EH K T S P EH R OW AA R M)
(AIR TO AIR SELECT SPARROW ARM)

(IX V EH L IY UW EH IY T TH R EH T IH N T IY S EH P T P R AA B AX B IH L UH T IY)
(IX V AE L IY UW EH IH T TH R EH T IH N T ER S EH P T P R AA B AX B IH L IX T IY)
(EVALUATE THREAT INTERCEPT PROBABILITY)

(AA IH D IY IH R K R AX F ZH T Y N AX K L AO CH T UW HH AH N D R AX D M AA IH L Z)
(AA IH D IY EH R K R AX F T T EH N AX K L AA K T UW HH AH N D R AX D M AA IH L Z)
(ID AIRCRAFT TEN OCLOCK TWO HUNDRED MILES)

(R AX IH D AA R EH N T ER T AX R G IX T S IH N T UW T R AE K F AA IH L)
(R EH IH D AA R EH N T ER T AA R G IX T S IH N T UW T R AE K F AA IH L)
(RADAR ENTER TARGETS INTO TRACK FILE)

(S EH L EH K SH SH AA AH N T ER M EH ZH Y ER F F OW R EH R T UW EH R TH R EH T)
(S EH L EH K T K AA UH N T ER M EH ZH Y ER Z F OW R EH R T UW EH R TH R EH T)
(SELECT COUNTERMEASURES FOR AIR TO AIR THREAT)

Appendix E: Lexicon constructed from Air Force study [LIZZ87]

```
(DEFVAR aa-cat '( ( 15 aa ih d ax n t ix f ix k eh ih sh ax n ( identification ) )
          ( 10 aa ih d eh n t ix f aa ih ( identify ) )
          ( 10 aa r d ah b l iy uw aa r ( rwr ) )
          ( 9 aa p er eh ih sh ax n l ( operational ) )
          ( 8 aa ih aa r eh s t iy ( irst ) )
          ( 8 aa r m ax m ax n t ( armament ) )
          ( 8 aa p t ix m aa ih z ( optimize ) )
          ( 7 aa p t ix m ax m ( optimum ) )
          ( 6 aa p t ix m l ( optimal ) )
          ( 6 aa p sh ax n z ( options ) )
          ( 6 aa r t iy b iy ( rtb ) )
          ( 5 aa p sh ax n ( option ) )
          ( 4 aa r m d ( armed ) )
          ( 4 aa r m z ( arms ) )
          ( 4 aa r v iy ( rv ) )
          ( 4 aa ih eh m ( im ) )
          ( 4 aa ih d iy ( id ) )
          ( 4 aa ih aa r ( ir ) )
          ( 3 aa ih v ( ive ) )
          ( 3 aa r m ( arm ) )
          ( 3 aa uh t ( out ) )
          ( 3 aa w er ( our ) )
          ( 3 aa p s ( ops ) )
          ( 2 aa n ( on ) )
          ( 2 aa r ( are ) )
          ( 2 aa ih ( i eye ) ) ) )

(DEFVAR ae-cat '( ( 8 ae k s eh p t ix d ( accepted ) )
          ( 8 ae k t ix v eh ih t ( activate ) )
          ( 7 ae k n aa l ix jh ( acknowledge ) )
          ( 7 ae l t ix t uw d ( altitude ) )
          ( 7 ae n ax l aa ih z ( analyze ) )
          ( 6 ae m r ae m z ( amraams ) )
          ( 6 ae s p eh k t ( aspect ) )
          ( 5 ae k sh ax n ( action ) )
          ( 5 ae k t ix v ( active ) )
          ( 5 ae m r ae m ( amraam ) )
          ( 5 ae l aa ih d ( allied ) )
          ( 3 ae n d ( and ) )
          ( 2 ae m ( am ) )
          ( 2 ae n ( an ) )
          ( 2 ae t ( at ) )
          ( 2 ae z ( as ) ) ) )

(DEFVAR ah-cat '( ( 8 ah p d eh ih dx ix d ( updated ) )
          ( 8 ah p d eh ih t ix d ( updated ) )
          ( 8 ah p d eh ih t ix ng ( updating ) )
          ( 6 ah p d eh ih t ( update ) )
          ( 3 ah dh er ( other ) )
          ( 2 ah p ( up ) ) ) )

(DEFVAR ao-cat '( ( 9 ao l t er n ix t ix v ( alternative ) )
          ( 8 ao l t er n ax t s ( alternates ) )
          ( 8 ao l t er n ih t s ( alternates ) )
          ( 8 ao t ax m ae t ix k ( automatic ) )
          ( 7 ao f eh n s ix v ( offensive ) )
          ( 7 ao l t er n ax t ( alternate ) )
          ( 7 ao l t er n ih t ( alternate ) )
          ( 5 ao f eh n s ( offense ) )
          ( 5 ao f s eh t ( offset ) )
```

```
        ( 4 ao l t er ( alter ) )
        ( 3 ao t ow ( auto ) )
        ( 2 ao f ( off ) )
        ( 2 ao l ( all ) ) ) )

(DEFVAR ax-cat '( ( 10 ax s aa ih n m ax n t s ( assignments ) )
        ( 9 ax k w ih p m ax n t ( equipment ) )
        ( 9 ax s aa ih n m ax n t ( assignment ) )
        ( 8 ax t eh m p t ix d ( attempted ) )
        ( 8 ax s eh s m ax n t ( assessment ) )
        ( 8 ax n ae l ix s ix s ( analysis ) )
        ( 8 ax k t ih v ix t iy ( activity ) )
        ( 8 ax k aa m p l ix sh ( accomplish ) )
        ( 8 ax v eh ih l ax b l ( available ) )
        ( 8 ax v ao ih d ax n s ( avoidance ) )
        ( 7 ax t eh m p t s ( attempts ) )
        ( 7 ax k s p r eh s ( express ) )
        ( 7 ax k s p eh n d ( expend ) )
        ( 7 ax k s p ae n d ( expand ) )
        ( 7 ax f eh n s ix v ( offensive ) )
        ( 7 ax d ih sh ax n l ( additional ) )
        ( 7 ax v eh ih zh ax n ( evasion ) )
        ( 7 ax v ao ih d ix ng ( avoiding ) )
        ( 6 ax t eh m p t ( attempt ) )
        ( 6 ax t ae k er z ( attackers ) )
        ( 6 ax n l aa r jh ( enlarge ) )
        ( 6 ax g eh n s t ( against ) )
        ( 6 ax b r eh s t ( abreast ) )
        ( 6 ax d v aa ih z ( advise ) )
        ( 6 ax r aa uh n d ( around ) )
        ( 5 ax n n ow n ( unknown ) )
        ( 5 ax k l aa k ( oclock ) )
        ( 5 ax f eh n s ( offense ) )
        ( 5 ax v ao ih d ( avoid ) )
        ( 5 ax s aa ih n ( assign ) )
        ( 5 ax b aa uh t ( about ) )
        ( 4 ax t ae k ( attack ) )
        ( 4 ax t ae ch ( attach ) )
        ( 4 ax s eh s ( assess ) )
        ( 4 ax hh eh d ( ahead ) )
        ( 2 ax v ( of ) )
        ( 2 ax s ( us ) ) ) )

(DEFVAR b-cat '( ( 8 b r eh ih k aa uh t ( breakout ) )
        ( 7 b ae n d ix t s ( bandits ) )
        ( 6 b iy v iy aa r ( bvr ) )
        ( 6 b ae n d ix t ( bandit ) )
        ( 6 b l aa ih n d ( blind ) )
        ( 6 b iy d iy eh ih ( bda ) )
        ( 6 b eh ih s ix z ( bases ) )
        ( 6 b eh ih r ix ng ( bearing ) )
        ( 5 b ow g iy z ( bogeys ) )
        ( 5 b l ow ah p ( blowup ) )
        ( 5 b ax g ih n ( begin ) )
        ( 5 b sh z er z ( buzzers ) )
        ( 5 b aa m er z ( bombers ) )
        ( 5 b r eh ih k ( break ) )
        ( 4 b r ih ng ( bring ) )
        ( 4 b ow g iy ( bogey ) )
        ( 4 b iy d iy ( bd ) )
        ( 4 b er s t ( burst ) )
        ( 4 b eh t er ( better ) )
        ( 4 b eh s t ( best ) )
```

```
        ( 4 b ae t l ( battle ) )
        ( 4 b ae n d ( band ) )
        ( 4 b aa m er ( bomber ) )
        ( 4 b aa k s ( box ) )
        ( 4 b eh ih s ( base ) )
        ( 3 b ow th ( both ) )
        ( 3 b ow r ( bore ) )
        ( 3 b iy m ( beam ) )
        ( 3 b ih t ( bit ) )
        ( 3 b ih g ( big ) )
        ( 3 b ah g ( bug ) )
        ( 3 b ae k ( back ) )
        ( 3 b aa ih ( by ) ) ) )

(DEFVAR eh-cat '( ( 7 eh eh ih n jh ix z ( changes ) )
        ( 5 eh eh ih n jh ( change ) )
        ( 4 eh aa p er ( chopper ) )
        ( 3 eh uw z ( choose ) )
        ( 3 eh iy f ( chief ) )
        ( 3 eh eh k ( check ) )
        ( 3 eh ae f ( chaff ) ) ) )

(DEFVAR d-cat '( ( 11 d ix s ix g n eh ih sh ax n ( designation ) )
        ( 9 d eh z ix g n eh ih t ( designate ) )
        ( 8 d ix t eh k sh ax n ( detection ) )
        ( 8 d ix f eh n s ix v ( defensive ) )
        ( 8 d ix s k r aa ih b ( describe ) )
        ( 7 d ix s p eh n s ( dispense ) )
        ( 7 d er eh k sh ax n ( direction ) )
        ( 7 d ao g f aa ih t ( dogfight ) )
        ( 7 d iy t eh ih l z ( details ) )
        ( 7 d ix t eh ih l z ( details ) )
        ( 7 d ix s p l eh ih ( display ) )
        ( 6 d ix f eh n d ( defend ) )
        ( 6 d ax f eh n s ( defense ) )
        ( 6 d ix f aa ih n ( define ) )
        ( 6 d aa ih v er t ( divert ) )
        ( 6 d iy t eh ih l ( detail ) )
        ( 6 d ix t eh ih l ( detail ) )
        ( 6 d eh ih n jh er ( danger ) )
        ( 6 d ix p l ao ih ( deploy ) )
        ( 5 d ix f iy t ( defeat ) )
        ( 5 d er eh k t ( direct ) )
        ( 5 d ax v er t ( divert ) )
        ( 5 d ae m ix jh ( damage ) )
        ( 5 d eh ih t ax ( data ) )
        ( 4 d uw ix ng ( doing ) )
        ( 4 d r aa p ( drop ) )
        ( 3 d ow p ( dope ) )
        ( 2 d uw ( do ) ) ) )

(DEFVAR dh-cat '( ( 3 dh ae t ( that ) )
        ( 3 dh eh m ( them ) )
        ( 3 dh eh r ( there their ) )
        ( 3 dh ow z ( those ) )
        ( 3 dh eh ih ( they ) )
        ( 2 dh ax ( the ) ) ) )


(DEFVAR eh-cat '( ( 11 eh k s p eh n d ax b l z ( expendables ) )
        ( 9 eh k s eh l er eh ih t ( accelerate ) )
        ( 9 eh ih v iy aa n ix k s ( avionics ) )
        ( 8 eh v r iy b ah d iy ( everybody ) )
        ( 8 eh k s eh k iy uw t ( execute ) )
        ( 8 eh r p l eh ih n z ( airplanes ) )
```

```
        ( 7 eh v r iy th ix ng ( everything ) )
        ( 7 eh r k r ax f t ( aircraft ) )
        ( 7 eh n v eh l ow p ( envelope ) )
        ( 7 eh r p l eh ih n ( airplane ) )
        ( 7 eh ih p iy eh k s ( apx ) )
        ( 7 eh ih ch eh s d iy ( hsd ) )
        ( 7 eh l ax m ax n t ( element ) )
        ( 6 eh s k ow r t ( escort ) )
        ( 6 eh s aa r eh m ( srm ) )
        ( 6 eh r b ow r n ( airborne ) )
        ( 6 eh m p l ao ih ( employ ) )
        ( 6 eh m aa r eh m ( mrm ) )
        ( 6 eh l aa r eh s ( lrs ) )
        ( 6 eh k s eh p t ( accept ) )
        ( 6 eh ih eh t iy iy ( hte ) )
        ( 5 eh ih t iy n ( eighteen ) )
        ( 5 eh ih r iy ax ( area ) )
        ( 5 eh n eh m iy ( enemy ) )
        ( 4 eh n t er ( enter ) )
        ( 4 eh s eh ih ( sa ) )
        ( 4 eh ih m z ( aims ) )
        ( 3 eh ih m ( aim ) )
        ( 3 eh ih d ( aid ) )
        ( 3 eh n iy ( any ) )
        ( 2 eh r ( air ) )
        ( 2 eh ih ( a ) ) ) )

(DEFVAR e-cat '( ( 3 er s t ( irst ) ) ) )

(DEFVAR f-cat '( ( 9 f ow r m eh ih sh ax n ( formation ) )
        ( 8 f r eh n d l iy z ( friendlies ) )
        ( 7 f r eh n d l iy ( friendly ) )
        ( 6 f ow r g eh t ( forget ) )
        ( 6 f aa l ow ix ng ( following ) )
        ( 6 f aa ih t er z ( fighters ) )
        ( 5 f r eh n d ( friend ) )
        ( 5 f r ah n t ( front ) )
        ( 5 f ow r t iy ( forty ) )
        ( 5 f l eh r z ( flares ) )
        ( 5 f ih f t iy ( fifty ) )
        ( 5 f l aa ih t ( flight ) )
        ( 5 f aa ih t er ( fighter ) )
        ( 5 f iy uw ch er ( future ) )
        ( 4 f l eh r ( flare ) )
        ( 4 f eh n s ( fence ) )
        ( 4 f ae s t ( fast ) )
        ( 4 f aa l ow ( follow ) )
        ( 4 f l aa ih ( fly ) )
        ( 4 f aa ih v ( five ) )
        ( 4 f aa ih t ( fight ) )
        ( 4 f aa ih r ( fire ) )
        ( 4 f aa ih l ( file ) )
        ( 4 f iy uw l ( fuel ) )
        ( 3 f uh l ( full ) )
        ( 3 f ow r ( four for ) )
        ( 3 f iy t ( feet ) )
        ( 2 f ow ( foe ) )
        ( 2 f er ( for ) ) ) )

(DEFVAR g-cat '( ( 7 g r ae n t ix d ( granted ) )
        ( 6 g r aa uh n d ( ground ) )
        ( 4 g r iy n ( green ) )
        ( 4 g ow ih ng ( going ) )
```

```
( 4 g ih m iy ( gimme ) )
( 4 g ah n z ( guns ) )
( 4 g aa d z ( gods ) )
( 4 g aa ih z ( guys ) )
( 3 g uh d ( good ) )
( 3 g iy r ( gear ) )
( 3 g ih v ( give ) )
( 3 g eh t ( get ) )
( 3 g ah n ( gun ) )
( 3 g ae s ( gas ) )
( 3 g aa t ( got ) )
( 2 g ow ( go ) ) ) )

(DEFVAR hh-cat '( ( 9 hh ow r ix z aa n t l ( horizontal ) )
( 9 hh eh l ax k aa p t er ( helicopter ) )
( 8 hh aa s t aa ih l z ( hostitiles ) )
( 8 hh ow m p l eh ih t ( homeplate ) )
( 7 hh ah n d r ax d ( hundred ) )
( 7 hh aa s t aa ih l ( hostile ) )
( 7 hh aa ih l aa ih t ( highlight ) )
( 6 hh ih s t er iy ( history ) )
( 6 hh aa s t l z ( hostitiles ) )
( 6 hh aa ih ax s t ( highest ) )
( 5 hh eh d ix ng ( heading ) )
( 5 hh aa s t l ( hostile ) )
( 4 hh ow l d ( hold ) )
( 4 hh iy t er ( heater ) )
( 4 hh iy l ow ( helo ) )
( 4 hh eh l p ( help ) )
( 4 hh aa ih d ( hide ) )
( 4 hh aa uh z ( hows ) )
( 3 hh uh k ( hook ) )
( 3 hh ow m ( home ) )
( 3 hh iy t ( heat ) )
( 3 hh ih m ( him ) )
( 3 hh ae v ( have ) )
( 3 hh ae d ( had ) )
( 3 hh aa t ( hot ) )
( 3 hh aa n ( hahn ) )
( 3 hh aa ih ( high ) )
( 3 hh aa uh ( how ) )
( 2 hh iy ( he ) ) ) )

(DEFVAR ih-cat '( ( 9 ih m p l ax m ax n t ( implement ) )
( 8 ih n t er s eh p t ( intercept ) )
( 6 ih ng g r ax s ( ingress ) )
( 6 ih n r aa uh t ( inroute ) )
( 5 ih n r uw t ( inroute ) )
( 5 ih n f r ax ( infra ) )
( 4 ih n t uw ( into ) )
( 4 ih n t ax ( into ) )
( 4 ih n f ow ( info ) )
( 2 ih z ( is ) )
( 2 ih t ( it ) )
( 2 ih n ( in ) ) ) )

(DEFVAR ix-cat '( ( 11 ix n s t r ah k sh ax n z ( instructions ) )
( 11 ix n t eh ih r ax g eh ih t ( interrogate ) )
( 10 ix ng g eh ih jh m ax n t ( engagement ) )
( 10 ix n f er m eh ih sh ix n ( information ) )
( 9 ix v ae l iy uw eh ih t ( evaluate ) )
( 8 ix n ih sh iy eh ih t ( initiate ) )
( 8 ix n d uw r ax n s ( endurance ) )
( 8 ix ng k aa uh n t er ( encounter ) )
( 7 ix v eh ih s ix v ( evasive ) )
```

```
        ( 7 ix ng g eh ih jh d ( enagaged ) )
        ( 6 ix n f ow r m ( inform ) )
        ( 6 ix m p ae k t ( impact ) )
        ( 6 ix ng g eh ih jh ( engage ) )
        ( 5 iy g r eh s ( egress ) )
        ( 5 iy s iy eh m ( ecm ) )
        ( 3 iy s t ( east ) ) ) )

(DEFVAR iy-cat '( ( 2 iy uw ( you ) ) ) )

(DEFVAR jh-cat '( ( 8 jh iy aa m ax t r iy ( geometry ) )
        ( 6 jh iy s iy aa ih ( gci ) )
        ( 5 jh ae m ix ng ( jamming ) )
        ( 5 jh ae m er z ( jammers ) )
        ( 4 jh ae m er ( jammer ) )
        ( 4 jh ao ih n ( join ) )
        ( 3 jh ae z ( jazz ) )
        ( 3 jh ae m ( jam ) )
        ( 3 jh eh ih ( j ) ) ) )

(DEFVAR k-cat '( ( 12 k aa uh n t er m eh zh y er z ( countermeasures ) )
        ( 11 k aa uh n t er m eh zh y er ( countermeasure ) )
        ( 11 k aa uh n er m eh zh y er z ( countermeasures ) )
        ( 10 k aa uh n er m eh zh y er ( countermeasure ) )
        ( 9 k r aa ih t iy r iy ax ( criteria ) )
        ( 9 k ae l k y l eh ih t ( calculate ) )
        ( 8 k ax n v er zh ax n ( conversion ) )
        ( 8 k ax n f ih g y er ( configure ) )
        ( 8 k ax n t ih n iy uw ( continue ) )
        ( 7 k l ow s ix s t ( closest ) )
        ( 7 k ax n t r ow l ( control ) )
        ( 7 k ax n s eh n t ( consent ) )
        ( 7 k ax n f er m d ( confirmed ) )
        ( 7 k ax m p l iy t ( complete ) )
        ( 6 k r ao s ix ng ( crossing ) )
        ( 6 k l ih zh ax n ( collision ) )
        ( 6 k ax n f er m ( confirm ) )
        ( 6 k ax m eh n s ( commence ) )
        ( 6 k ax m ae n d ( command ) )
        ( 6 k aa m b ae t ( combat ) )
        ( 6 k aa uh n t er ( counter ) )
        ( 5 k l ow zh er ( closure ) )
        ( 5 k l ow s er ( closer ) )
        ( 5 k l iy r d ( cleared ) )
        ( 5 k ax m ih t ( commit ) )
        ( 5 k l aa ih m ( climb ) )
        ( 5 k aa ih n d ( kind ) )
        ( 5 k aa uh n t ( count ) )
        ( 4 k w ih k ( quick ) )
        ( 4 k r uw z ( cruise ) )
        ( 4 k r ao s ( cross ) )
        ( 4 k ow r s ( course ) )
        ( 4 k l ow z ( close ) )
        ( 4 k l iy r ( clear ) )
        ( 4 k ih l er ( killer ) )
        ( 3 k iy p ( keep ) )
        ( 3 k ih l ( kill ) )
        ( 3 k ao l ( call ) )
        ( 3 k ah m ( come ) )
        ( 3 k ae n ( can ) )
        ( 3 k iy uw ( cue ) ) ) )

(DEFVAR l-cat '( ( 8 l ow k eh ih sh ax n ( location ) )
        ( 5 l aa ih m ax ( lima ) )
        ( 4 l iy th l ( lethal ) )
```

```
        ( 4 l iy m ax ( lima ) )
        ( 4 l iy d er ( leader ) )
        ( 4 l ih ng k ( link ) )
        ( 4 l eh v l ( level ) )
        ( 4 l eh t s ( lets ) )
        ( 4 l eh f t ( left ) )
        ( 4 l ao n sh ( launch ) )
        ( 4 l aa k t ( locked ) )
        ( 4 l aa ih n ( line ) )
        ( 4 l aa ih k ( like ) )
        ( 3 l uh k ( look ) )
        ( 3 l ow d ( load ) )
        ( 3 l iy n ( lean ) )
        ( 3 l iy d ( lead ) )
        ( 3 l eh t ( let ) )
        ( 3 l ao ng ( long ) )
        ( 3 l aa k ( lock ) )
        ( 2 l ow ( low ) ) ) )

(DEFVAR m-cat '( ( 7 m ah l t ix p l ( multiple ) )
        ( 6 m iy d iy ax m ( medium ) )
        ( 6 m eh m b er z ( members ) )
        ( 6 m ax n uw v er ( maneuver ) )
        ( 6 m ae g n ax m ( magnum ) )
        ( 6 m aa n ix t er ( monitor ) )
        ( 6 m ae n iy uw l ( manual ) )
        ( 5 m ah dh er z ( mothers ) )
        ( 5 m ih sh ax n ( mission ) )
        ( 5 m ih s l z ( missiles ) )
        ( 5 m eh s ix jh ( message ) )
        ( 5 m ae s t er ( master ) )
        ( 5 m aa ih l z ( miles ) )
        ( 4 m ih s l ( missile ) )
        ( 4 m ae k s ( max ) )
        ( 4 m aa ih n ( mein ) )
        ( 4 m eh ih n ( main ) )
        ( 3 m uw v ( move ) )
        ( 3 m ow r ( more ) )
        ( 3 m ow d ( mode ) )
        ( 3 m ih g ( mig ) )
        ( 3 m ah d ( mud ) )
        ( 3 m ae p ( map ) )
        ( 3 m ae n ( man ) )
        ( 3 m aa k ( mach ) )
        ( 3 m aa ih ( my ) )
        ( 2 m iy ( me ) ) ) )

(DEFVAR n-cat '( ( 10 n ae v ix g eh ih sh ax n ( navigation ) )
        ( 7 n eh g ix t ix v ( negative ) )
        ( 7 n ow t ax f aa ih ( notify ) )
        ( 6 n iy r ax s t ( nearest ) )
        ( 6 n ah m b er z ( numbers ) )
        ( 6 n aa ih n t iy ( ninety ) )
        ( 5 n uw ax s t ( newest ) )
        ( 5 n ah m b er ( number ) )
        ( 4 n ow r th ( north ) )
        ( 4 n eh r ow ( narrow ) )
        ( 4 n aa ih n ( nine ) )
        ( 3 n ow z ( nose ) )
        ( 3 n eh t ( net ) )
        ( 3 n ae v ( nav ) )
        ( 3 n aa n ( non ) )
        ( 3 n aa uh ( now ) )
        ( 2 n uw ( new ) )
        ( 2 n ow ( know ) ) ) )
```

```
(DEFVAR ow-cat '( ( 7 ow r d n ax n s ( ordnance ) )
                 ( 6 ow v er v iy nw ( overview ) )
                 ( 5 ow v er ao l ( overall ) )
                 ( 4 ow n l iy ( only ) )
                 ( 4 ow k eh ih ( ok ) )
                 ( 2 ow r ( or ) ) ) )

(DEFVAR p-cat  '( ( 12 p r aa ih ow r ix t aa ih z d ( prioritized ) )
                 ( 12 p r eh s ax n t eh ih sh ax n ( presentation ) )
                 ( 11 p r aa b ax b ih l ix t iy ( probability ) )
                 ( 11 p r aa ih ow r ix t aa ih z ( prioritize ) )
                 ( 10 p r aa ih ow r ix t iy z ( priorities ) )
                 ( 9 p er f ow r m ax n s ( performance ) )
                 ( 9 p r aa ih ao r ix t iy ( priority ) )
                 ( 8 p l ae t f ow r m ( platform ) )
                 ( 8 p er ae m ax t er z ( parameters ) )
                 ( 8 p r ow f aa ih l z ( profiles ) )
                 ( 8 p r aa ih m eh r iy ( primary ) )
                 ( 7 p r ow g r ae m ( program ) )
                 ( 7 p r eh s ax n t ( present ) )
                 ( 7 p r aa jh eh k t ( project ) )
                 ( 7 p ax s ih sh ax n ( position ) )
                 ( 7 p r ow f aa ih l ( profile ) )
                 ( 6 p r iy p eh r ( prepare ) )
                 ( 7 p r aa ih m er iy ( primary ) )
                 ( 6 p r ax s iy d ( proceed ) )
                 ( 6 p ih jh ax n z ( pigeons ) )
                 ( 6 p er f ow r m ( perform ) )
                 ( 6 p l eh ih n z ( planes ) )
                 ( 5 p ih n s er ( pincer ) )
                 ( 5 p ih k ch er ( picture ) )
                 ( 5 p eh r ax t ( parrot ) )
                 ( 5 p ae s ix v ( passive ) )
                 ( 5 p ae k ix jh ( package ) )
                 ( 5 p aa s ix t ( posit ) )
                 ( 5 p l eh ih t ( plate ) )
                 ( 5 p iy k eh ih ( pk ) )
                 ( 5 p ao ih n t ( point ) )
                 ( 4 p r eh s ( press ) )
                 ( 4 p l ae n ( plan ) )
                 ( 4 p l aa t ( plot ) )
                 ( 4 p ih n sh ( pinch ) )
                 ( 4 p aa d z ( pods ) )
                 ( 3 p uh t ( put ) )
                 ( 3 p ih n ( pin ) )
                 ( 3 p ae th ( path ) )
                 ( 3 p ae s ( pass ) )
                 ( 3 p aa d ( pod ) ) ) )

(DEFVAR r-cat  '( ( 11 r iy ix s aa ih n  m ax n t ( reassignment ) )
                 ( 11 r iy k ae k iy uw l eh ih t ( recalculate ) )
                 ( 10 r ix t aa r g ih t ix ng ( retargeting ) )
                 ( 10 r ix k ix n f ih g y er ( reconfigure ) )
                 ( 8 r iy t aa r g ax t ( retarget ) )
                 ( 8 r ae m s t aa ih n ( ramstein ) )
                 ( 7 r iy k w eh s t ( request ) )
                 ( 7 r ix k w eh s t ( request ) )
                 ( 7 r ix k ah v er iy ( recovery ) )
                 ( 7 r eh l ix t ix v ( relative ) )
                 ( 7 r aa n d ix v nw ( rendezvous ) )
                 ( 7 r iy ix s aa ih n ( reassign ) )
                 ( 7 r eh ih d iy ax s ( radius ) )
                 ( 6 r ix z ow r t ( resort ) )
                 ( 6 r ix p ow r t ( report ) )
                 ( 6 r ix k ah v er ( recover ) )
```

```
( 6 r eh ih d iy ow ( radio ) )
( 6 r eh ih d aa r ( radar ) )
( 6 r iy jh ao ih n ( rejoin ) )
( 6 r iy r aa uh t ( reroute ) )
( 6 r aa uh t ix ng ( routing ) )
( 5 r uw t ix ng ( routing ) )
( 5 r iy t er n ( return ) )
( 5 r iy r uw t ( reroute ) )
( 5 r iy d ix ng ( reading ) )
( 5 r iy l eh ih ( relay ) )
( 5 r ix t er n ( return ) )
( 5 r ix p iy t ( repeat ) )
( 5 r ix m ow d ( remode ) )
( 5 r iy l eh ih ( relay ) )
( 5 r eh ih n jh ( range ) )
( 5 r aa uh t s ( routes ) )
( 4 r uw t s ( routes ) )
( 4 r eh s t ( rest ) )
( 4 r eh d iy ( ready ) )
( 4 r aa ih t ( right ) )
( 4 r aa ih n ( rhein ) )
( 4 r eh ih n ( rain ) )
( 4 r eh ih d ( raid ) )
( 4 r aa uh t ( route ) )
( 3 r uw t ( route ) )
( 3 r ih ng ( ring ) )
( 3 r eh d ( red ) )
( 3 r ae m ( ram ) )
( 2 r ao ( raw ) ) ) )

(DEFVAR s-cat ' ( ( 14 s aa ih m l t eh ih n iy ax s l iy ( simultaneously ) )
( 11 s aa ih m l t ae n iy ax s ( simultaneous ) )
( 10 s aa ih d w aa ih n d er ( sidewinder ) )
( 8 s ax l eh k t ix d ( selected ) )
( 8 s ax l eh k sh ax n ( selection ) )
( 8 s p ax s ih f ix k ( specific ) )
( 8 s t r aa ih k er z ( strikers ) )
( 8 s t ae n d b aa ih ( standby ) )
( 8 s p eh s ax f aa ih ( specify ) )
( 8 s p aa t l aa ih t ( spotlight ) )
( 8 s ih ch uw eh ih sh n ( situation ) )
( 7 s ih s t ax m z ( systems ) )
( 7 s ih g n ax ch er ( signature ) )
( 7 s eh p ax r ax t ( separate ) )
( 7 s eh k ax n d z ( seconds ) )
( 7 s ax g jh eh s t ( suggest ) )
( 7 s t r aa ih k er ( striker ) )
( 7 s t eh ih t ax s ( status ) )
( 7 s eh ih f ax s t ( safest ) )
( 7 s eh p er eh ih t ( separate ) )
( 6 s uw t ax b l ( suitable ) )
( 6 s t r ih p t ( stripped ) )
( 6 s t iy r ix ng ( steering ) )
( 6 s p aa r k l ( sparlkle ) )
( 6 s ow r t ix d ( sorted ) )
( 6 s k eh d y l ( schedule ) )
( 6 s k ae n er z ( scanners ) )
( 6 s ih s t ax m ( system ) )
( 6 s ih k s t iy ( sixty ) )
( 6 s eh p r ix t ( separate ) )
( 6 s t r aa ih k ( strike ) )
( 6 s aa ih d ix d ( sided ) )
( 6 s n eh ih k s ( snakes ) )
( 6 s eh l eh k t ( select ) )
( 5 s t ow r z ( stores ) )
```

```
( 5 s t ae t s ( stats ) )
( 5 s t aa r t ( start ) )
( 5 s p l ih t ( split ) )
( 5 s p eh r ow ( sparrow ) )
( 5 s k r iy n ( screen ) )
( 5 s ih ng g l ( single ) )
( 5 s ih g n l ( signal ) )
( 5 s eh v ax n ( seven ) )
( 5 s eh t ix ng ( setting ) )
( 5 s eh n s er ( sensor ) )
( 5 s ah k er z ( suckers ) )
( 5 s ae m p l ( sample ) )
( 5 s ae l v ow ( salvo ) )
( 5 s t eh ih t ( state ) )
( 5 s k eh ih l ( scale ) )
( 5 s eh ih b er ( saber ) )
( 4 s w iy p ( sweep ) )
( 4 s w ih ng ( swing ) )
( 4 s w ih ch ( switch ) )
( 4 s w aa p ( swap ) )
( 4 s t ow r ( store ) )
( 4 s t iy r ( steer ) )
( 4 s p iy k ( speak ) )
( 4 s p iy d ( speed ) )
( 4 s ow r t ( sort ) )
( 4 s n ae p ( snap ) )
( 4 s k ow p ( scope ) )
( 4 s k ae n ( scan ) )
( 4 s iy t er ( seater ) )
( 4 s ih k s ( six ) )
( 4 s eh n d ( send ) )
( 4 s eh l f ( self ) )
( 4 s aa ih t ( site ) )
( 4 s eh ih f ( safe ) )
( 3 s er ch ( search ) )
( 3 s eh t ( set ) )
( 3 s ah b ( sub ) )
( 3 s ae m ( sam ) )
( 3 s eh ih ( say ) )
( 2 s iy ( see c ) ) ) )

(DEFVAR sh-cat '( ( 4 sh ow r t ( short ) )
                  ( 3 sh aa t ( shot ) )
                  ( 3 sh eh r ( share ) )
                  ( 3 sh er k ( shirk ) )
                  ( 3 sh uw t ( shoot ) )
                  ( 2 sh ow ( show ) ) ) )

(DEFVAR t-cat  '( ( 10 t iy d ah b l iy uw eh s ( tws ) )
                  ( 8 t iy eh f t iy eh ih ( tfta ) )
                  ( 8 t r ax n z m ih t ( transmit ) )
                  ( 8 t aa r g ix t ix ng ( targeting ) )
                  ( 8 t aa r g ix t ix d ( targeted ) )
                  ( 7 t r ae n s f er ( transfer ) )
                  ( 7 t ae k t ix k s ( tactics ) )
                  ( 7 t ae k t ix k l ( tactical ) )
                  ( 7 t aa r g ix t s ( targets ) )
                  ( 6 t w eh n t iy ( twenty ) )
                  ( 6 t r ae k ix ng ( tracking ) )
                  ( 6 t iy eh f aa r ( tfr ) )
                  ( 6 t ae ng k er z ( tankers ) )
                  ( 6 t ae k t ix k ( tactic ) )
                  ( 6 t aa r g ix t ( target ) )
                  ( 6 t r eh ih l er ( trailer ) )
                  ( 5 t w eh l v ( twelve ) )
```

```
        ( 5 t r ah b l ( trouble ) )
        ( 5 t r eh ih l ( trail ) )
        ( 5 t er eh ih n ( terrain ) )
        ( 4 t w aa z ( twos ) )
        ( 4 t r ae k ( track ) )
        ( 4 t iy eh f ( tf ) )
        ( 4 t eh s t ( test ) )
        ( 4 t aa ih p ( type ) )
        ( 4 t aa ih m ( time ) )
        ( 4 t eh ih k ( take ) )
        ( 3 t ow t ( tot ) )
        ( 3 t er n ( turn ) )
        ( 3 t eh n ( ten ) )
        ( 3 t eh l ( tell ) )
        ( 2 t uw ( two to ) )
        ( 2 t uh ( two to ) )
        ( 2 t ix ( to ) )
        ( 2 t ax ( to ) ) ) ) )

(DEFVAR th-cat '( ( 5 th r eh t s ( threats ) )
        ( 4 th er t iy ( thirty ) )
        ( 4 th r eh t ( threat ) )
        ( 3 th r iy ( three ) )
        ( 3 th r uw ( through ) ) ) ) )


(DEFVAR v-cat  '( ( 8 v er b ax l aa ih z ( verbalize ) )
        ( 7 v iy s ah b s iy ( vsubc ) )
        ( 6 v eh k t er z ( vectors ) )
        ( 5 v eh k t er ( vector ) )
        ( 5 v ih zh uw l ( visual ) )
        ( 4 v aa ih d ( vid ) )
        ( 3 v iy uw ( view ) ) ) )

(DEFVAR w-cat  '( ( 8 w ih ng g m ax n z ( wingmans ) )
        ( 7 w ih ng g m eh n ( wingmen ) )
        ( 7 w ih ng g m ax n ( wingman ) )
        ( 6 w eh p ax n z ( weapons ) )
        ( 5 w ih n d ow ( window ) )
        ( 5 w ih l k ow ( wilco ) )
        ( 5 w er k ix ng ( working ) )
        ( 5 w eh p ax n ( weapon ) )
        ( 4 w er s t ( worst ) )
        ( 4 w eh r z ( wheres ) )
        ( 4 w ax t s ( whats ) )
        ( 4 w ah n z ( ones ) )
        ( 4 w aa n t ( want ) )
        ( 4 w aa n ax ( wanna ) )
        ( 4 w aa ih l ( while ) )
        ( 4 w aa ih d ( wide ) )
        ( 3 w ih th ( with ) )
        ( 3 w ih ng ( wing ) )
        ( 3 w ih l ( will ) )
        ( 3 w ih ch ( which ) )
        ( 3 w eh r ( where ) )
        ( 3 w eh n ( when ) )
        ( 3 w eh l ( well ) )
        ( 3 w ax t ( what ) )
        ( 3 w ah n ( one ) )
        ( 3 w eh ih ( way ) )
        ( 2 w iy ( we ) )
        ( 2 w er ( were ) ) ) )

(DEFVAR y-cat  '( ( 4 y aa uh v ( youve ) )
        ( 3 y eh s ( yes ) )
```

```
        ( 3 y ow r ( your ) )
        ( 2 y er ( your ) ) ) )

(DEFVAR z-cat '( ( 4 z iy r ow ( zero ) )
        ( 3 z ae p ( zap ) )
        ( 3 z ow n ( zone ) )
        ( 3 z uw m ( zoom ) ) ) )
```

BIBLIOGRAPHY

[BAKE75] Baker, J. K., "The DRAGON System - An Overview," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 23, Feb 1975, pp. 24-29.

[CHAR86] Charot, F., "Systolic Architectures for Connected Speech Recognition," *EEE Transactions on Acoustices, Speech, and Signal Processing*, Vol. 34, August 1986, pp. 765-769.

[COHE75] Cohen, P. S. and Mercer, R. L., "The Phonological Component of an Automatic Speech-Recognition System," *Speech Recognition: Invited Papers Of The 1974 IEEE Symposium*, Reddy, D.R. ed., Academic Press, New York, 1975, pp. 275-320.

[COLE80] Cole, R. A. and Jakimik, J., "A Model of Speech Perception," *Production and Perception Of Fluent Speech*, Erlbaum ed., Hillsdale, N.J., 1980, p. 139.

[ERIC83] Erickson, B. W. and Sellers, P. H., "Recognition of Patterns in Genetic Sequences," *Time Warps, String Edits, and Macromolecules: The Theory And Practice Of Sequence Comparison*, Sankoff, D. and Kruskal, J. B. Eds., Addison-Wesley, Reading, Ma., 1983, pp. 55-68.

[ERNS69] Ernst, G. and Newell, Allen,*GPS: A Case Study in Generality and Problem Solving*, Academic Press, New York 1969.

[HILL87] Hillenbrand, J. and Gayvert, R. T., "Speaker-Independent Vowel Classification Based on Fundamental Frequency and Format Frequencies," *Journal of the Acoustical Society of America*, Spring 1987, Vol. 81 (Suppl. 1), S93 (A).

[HILL88] Hillenbrand, J. and Gayvert, R. T., Advisement and Personal Conversation, May 30, 1988.

[HUTT84] Huttenlocher, D., and Zue, V., "A Model of Lexical Access from Partial Phonetic Information," *Proceed. ICASSP 1984*, #CH1945- 5/84/0000-0277.

[HYDE72] Hyde, S. R., "Automatic Speech Recognition: A Critical Survey and Discussion of the Literature," *Human Communication: A Unified View*, David Jr., E.E., and Denes, P.B., Eds., McGraw-Hill, New York, 1972, pp. 399-438.

[ITAK75] Itakura, F., "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Transactions Acoustical, Speech, Signal Processing*, Vol. ASSP-23, Feb. 1975, pp.67-72.

[JELI74] Jelinek, F., Bahl, L. R., and Mercer, R. L., "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," *IEEE Symposium on Speech Recognition*, April 15-19, 1974, pp. 255-260.

[JELI76] Jelinek, F., "Continuous Speech Recognition by Statistical Methods," *Proceedings IEEE*, Vol. 64, 1976, pp. 532-556.

[KLAT75] Klatt, D. H., "Word Verification in a Speech Understanding System," *Speech Recognition: Invited Papers Of The 1974 IEEE Symposium*, Reddy, D.R. ed., Academic Press, New York, 1975, pp. 321-341.

[KLAT76] Klatt, D., "Review of the ARPA Speech Understanding Project," *Journal of the Acoustic Society of America*, Vol. 62, 1976, pp. 1345-1366.

[KRUS83] Kruskal, J. B., "Overview of Sequence Comparison," *Time Warps, String Edits, and Macromolecules: The Theory And Practice Of Sequence Comparison*, Sankoff, D. and Kruskal, J.B. Eds., Addison-Wesley, Reading, Ma., 1983, pp. 1-40.

[KRUS83a] Kruskal, J. B. and Sankoff, D. "An Anthology of Algorithms and Concepts For Sequence Comparison," *Time Warps, String Edits, and Macromolecules: The Theory And Practice Of Sequence Comparison*, Sankoff, D. and Kruskal, J. B. Eds., Addison-Wesley, Reading, Ma., 1983, pp. 265-311.

[LESS75] Lesser, V. R., Fennell, R. D., Erman, L. D., and Reddy, D. R., "Organization of the Hearsay II Speech Understanding System," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, Feb. 1975, pp. 11-24.

[LEVI83] Levinson, S. E., Rabiner, L. R., and Sondhi, M. M., "Speaker Independent Isolated Digit Recognition Using Hidden Markov Models," *Proceedings of ICASSP Boston*, 1983, Vol. 3, pp.1049-1052.

[LEVI85] Levinson, S. E., "Structural Methods in Automatic Speech Recognition," *Proceedings of the IEEE*, Vol. 73 No. 11, November 1985, pp. 1625-1650.

[LEVI87] Levinson, S. E., " Continuous Speech Recognition by Means of Acoustic/ Phonetic Classification Obtained from a Hidden Markov Model," *Proceedings of ICASSP*, April 1987, Vol. 1, pp.93-95.

[LIZZ87] Lizza, Capt. G., Munger, M., Small, Capt. R., Feitshans, G., and Detro, S., "A Cockpit Natural Language Study - Data Collection and Initial Data Analysis," Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, April 1987, Doc. # AFWL-TR-87-3003.

[LOWE80] Lowerre, B. and Reddy, D. R., "The Harpy Speech Understanding System," *Trends In Speech Recognition*, Lea, W.A. ed., Prentice Hall, Englewood Cliffs, N.J., 1980, pp. 101-124.

[MART75] Martin, T. B., "Applications of Limited Vocabulary Recognition Systems," *Speech Recognition: Invited Papers Of The IEEE Symposium*, Reddy, D.R. ed., Academic Press, New York, 1975, pp. 55-71.

[MART87] Martin, G. L., "The Utility of Speech Input in User-Computer Interfaces," *MCC Technical Report Number HI-021-87*, Microelectronics and Computer Technology Corp., December 1987, pp. 1-32.

[MILL55] Miller, G. and Nicely, P., "An Analysis of Perceptual Confusions Among Some English Consonants," *Journal of the Acoustic Society of America*, Vol. 27, 1955, pp. 338-352.

[MYER81] Myers, C. S. and Rabiner, L. R., "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 29, April 1981, pp. 286-297.

[NEY84] Ney, Hermann, "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition," *IEEE Transactions on Acoustics Speech, and Signal Processing*, Vol. 32, April 1984, pp. 263-271.

[OSKI75] Oskika, B. T., Zue, V. W., Weeks, R., Neu, H., and Aurbach J., "The Role of Phonological Rules in Speech Understanding Research," *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-23, No. 1, Feb. 1975, pp. 104-112.

[PARS86] Parsons, T., *Voice And Speech Processing*, McGraw-Hill, New York, 1986, pp. 291-331.

[PETE52] Peterson, G., and Barney, H., "Control Methods Used in a Study of the Vowels," *Journal of the Acoustic Society of America*, Vol. 24, 1952, pp. 182-184.

[PICO86] Picone, J., Goudie-Marshall, K., Doddington, G. and Fisher, W., "Automatic Text Alignment for Speech System Evaluation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 34, August 1986, pp. 780-784.

[RABI86] Rabiner, L. R. and Juang, B. H., "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, January 1986, pp 4-16.

[REDD76] Reddy, D. R., "Speech Recognition by Machine: A Review," *Proceedings of the IEEE*, Vol. 64, April 1976, pp. 501-531.

[RUDN87] Rudnicky, A., Baumeister, L., DeGraaf, K., and Lehmann, E., "The Lexical Access Component of the CMU Continuous System," *Proceedings: ICASSP 87*, Vol. 1, April 1987, pp. 376-379.

[SAKO78] Sakoe, H. and Chiba, S., "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*,

Vol. 26, February 1978, pp. 43-49.

[SAKO79] Sakoe, H., "Two-Level DP-Matching -- A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 27, December 1979, pp. 588-595.

[SHEP80] Shepard, R., "Multidimensional Scaling, Tree-Fitting, and Clustering," *Science*, Vol. 210, October 1980, pp. 390-398.

[SHIP82] Shipman, D. W. and Zue, V. W., "Properties of Large Lexicons: Implications for Advanced Isolated Word Recognition Systems," *ICASSP 1982 Proceedings*, pp. 546-549.

[SHOU80] Shoup, J. E., "Phonological Aspects of Speech Recognition," *Trends In Speech Recognition*, Lea, W.A. ed., Prentice Hall, Englewood Cliffs, NJ., 1980, pp. 101-124.

[SMIT77] Smith, A. R., "Word Hypothesization for Large-Vocabulary Speech Understanding Systems," PhD Thesis: Carnegie-Mellon University, October 1977.

[SMIT80] Smith, A. R. and Sambur, M. R., "Hypothesizing and Verifying Words for Speech Recognition," *Trends In Speech Recognition*, Lea, W.A. ed., Prentice Hall, Englewood Cliffs, NJ., 1980, pp. 101-124.

[STEE84] Steele Jr., G. L., *COMMON LISP: The Language*, Digital Press, Hanover, Ma., 1984.

[WAIB81] Waibel, A. and Yegnanarayana, B., "Comparative Study of Nonlinear Time Warping Techniques in Isolated Word Speech Recognition Systems," Research Paper: Carnegie-Mellon University, CMU-CS-81-125, June 1981.

[WATA86] Watari, M., "New DP Matching Algorithms for Connected Word Recognition," *Proceedings of ICASSP Tokyo*, 1986, Vol.2, pp.1113-1116.

[WATE86] Waterman, D. A., *A Guide To Expert Systems*, Addison-Wesley, Reading, Ma., 1986, pp. 1-26.

[WHIT75] White, G. M. and Neely, R. B., "Speech Recognition Experiments with Linear Prediction, Bandpass Filtering, and Dynamic Programming," *Proceedings 2nd USA-Japan Computer Conf.*, Tokyo, Japan, August 1975.

[WHIT76] White, G. M., "Speech Recognition: A Tutorial Overview," *COMPUTER*, Vol. 9, May 1976, pp. 40-53.

[WINS84] Winston, P. H., *Artificial Intelligence*, Addison-Wesley, Reading, Ma., 1984, pp. 1-42, 87-135, 253-290.

[WITT82] Witten, I. H., *Principles Of Computer Speech*, Academic Press, London, 1982.

[WOLF77] Wolf, J. J. and Woods, W. A., "The HWIM Speech Understanding System," *IEEE Internat. Conf. Record on Acoustics, Speech, and Signal Process.*, May 1977, pp. 784-787.

[WOOD75] Woods, W. A., "Motivation and Overview of SPEECHLIS: An Experimental Prototype for Speech Understanding Research," *IEEE Trans. Acoustics, Speech, and Signal Process.*, Vol. ASSP-23, Feb. 1975, pp. 2-10.

[ZUE80] Zue, V. and Schwartz, R. M., "Acoustic Processing and Phonetic Analysis," *Trends In Speech Recognition*, Lea, W.A. ed., Prentice Hall, Englewood Cliffs, NJ., 1980, pp. 101-124.