

Computer Recognition Of Human Speech

Mathew Joseph Palakal

A Thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada.

October 1983

© Mathew Joseph Palakal, 1983

ABSTRACT

COMPUTER RECOGNITION OF HUMAN SPEECH

Mathew Joseph Palakal

Speech is the most efficient and convenient means for communicating information among humans. If computers could be given the ability for voice communication, their value and ease of use for human would increase. To develop economically, machines that could talk and listen, need more intensive research.

First part of this thesis is a detailed study of the problems associated with isolated-word recognition systems. A speaker-independent isolated-word recognition system was developed and implemented on a microprocessor based system. Speaker normalization was achieved by keeping multiple reference templates which could accommodate variations of many speakers. A clustering technique was used to create multiple templates.

In the second part, a new solution for accessing a large lexicon in speaker-independent continuous speech recognition systems is proposed and successfully

implemented.

The words of a lexicon and their relations with syllables, acoustics, and prosodic cues are represented by a network. A lexical access tree is built during the learning stage which would access a set of words from the lexicon when some Sufficient Conditions are detected in the data.

An experiment was conducted to test the percentage of words accessed from the lexicon and to verify speaker-independency. Only 2.3% of the words were accessed from a lexicon of 9000 words.

ACKNOWLEDGEMENTS

I am greatly indebted to my supervisor, Prof. Renato De Mori, for his expert guidance, advice, and financial support during the course of this research.

I also wish to express my sincere gratitude to Prof. C. Y. Suen for his kindness and guidance both financially and otherwise throughout the course my research. I thank him also for providing the lexicon and the software for syllabification rules.

I wish to thank Fr. James MacDonald for his unfailing support during my stay in Canada.

Special thanks to Miss Swetha Ramaswamy for proof reading this thesis.

I wish to thank Miss Yu Mong and Mr. Michael Yu for their assistance.

I also wish to thank my family in India and all my dear friends in Germany and in Canada for their support and kindness.

CONTENTS

ABSTRACT iii

ACKNOWLEDGMENTS v

I. INTRODUCTION 1

 1.1 Speech Production 1

 1.2 Linguistic Aspect of Speech 3

 1.3 Acoustic Characteristics of Phonemes 5

 1.4 Speech Recognition by Computer 16

 1.5 Speech Recognition Systems:
 Past and Present 20

II. A SPEAKER-INDEPENDENT ISOLATED WORD RECOGNITION
SYSTEM 24

 2.1 Hardware Organization of the System 24

 2.2 Preprocessing 26

 2.3 Test Data Collection 29

 2.4 Recognition Procedure 32

 2.5 Results and Conclusions 39

III. CONTINUOUS SPEECH RECOGNITION 42

 3.1 Continuous Speech Recognition Systems 43

 3.2 Expert Systems in Speech Recognition 46

 3.3 Representation of Expert's Knowledge 51

 3.4 Interaction of Experts to Decode Speech 57

IV. PROBLEM OF ACCESSING A LARGE LEXICON:
A NEW SOLUTION 65

 4.1 Lexical Representation and Lexical
 Accessing 65

 4.2 Syllabification using Dynamic
 Programming 67

 4.3 Lexical Access 77

 4.4 Organization of Lexical Knowledge 82

V. SYSTEM PERFORMANCE AND EVALUATION 91

 5.1 Experiments on Lexical Accessing 91

 5.2 Conclusion 100

REFERENCES 102

APPENDIX 107

LIST OF FIGURES

1.1 Organs of Speech Production	2
1.2 Tree Representation of Phonemes	4
1.3 Generation of Vocalic Sounds	7
1.4 Place of Articulation of Stop Consonants	10
1.5 Generation of Fricative Sounds	12
1.6 Classification of Nasals and Glides	14
1.7 Passive Model Speech Recognition Systems	17
1.8 Active Model Speech Recognition System	19
2.1 Schematic Diagram of the Speech Board	25
2.2 Preprocessing and Recognition	27
2.3 Clustering Algorithm	31
2.5 Peak Labelling Algorithm	35
2.6 Classification Algorithm	36
3.1 Expert System Society for Speech Decoding	50
3.2 Frame Structure	56
3.3 Generating Acoustic Cues from Energy Signal	59
4.1 DP-Matching for Time Warping	68
4.2 Example of Deletion, Insertion, Substitution	72
4.3 DP-Matching with added Weight Function	74
4.4 Syllabification using DP-Matching	75
4.5 Lexical Access Model	78
4.6 Model of Lexical Access Tree	80
4.7 Algorithm to Access Large Lexicon	89
5.1 Algorithm for Sufficient Conditions	92
5.2a Output of the Expert, PPFD	95

5.2b Section of the Lexical Access Tree	96
5.3 Graph of Tree Learning %	98
5.4 Graph of Tree Updates	99

LIST OF TABLES

1.1 Evaluation of Word Recognition Systems	22
3.1 Rules of the frame-structure grammar	53
3.2 The Primary Acoustics Cues	60
3.3 Primary Phonetic Cues	62

Chapter I

INTRODUCTION

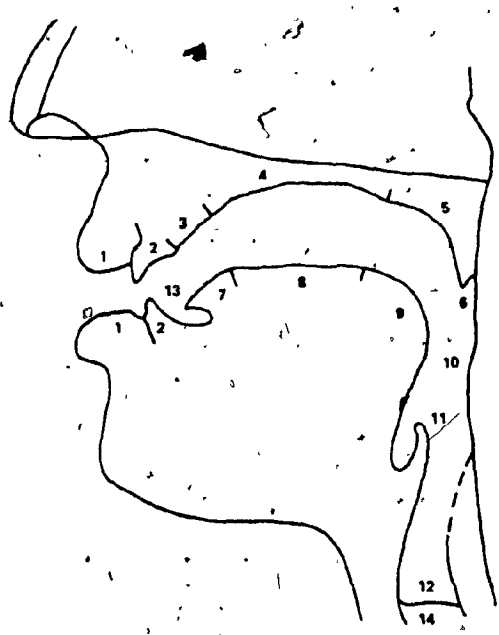
One of the most predominant characteristics which distinguishes human beings from animals is the ability to speak. Even though human beings have numerous means of communication, such as writing, Morse Code, and sign languages, speech remains the most powerful.

Speech processing on computers has been of interest to researchers since the early 1950's. Speech processing is composed of two basic branches, namely, Speech Synthesis and Speech Recognition. Although there has been considerable development in the speech synthesis field, little progress has been made in speech recognition because of the complexity of speech signals. For instance, there is a great extent of variation in the same spoken word from speaker to speaker. In the following section we shall examine how speech is produced by human beings.

1.1 Speech Production

Human speech is produced by a physical system consisting of four main parts: lungs, vocal tract, nasal

Fig 1.1 Organs of Speech Production



- | | | |
|----------------|------------------|-------------------|
| 1. Lips | 5. Velum | 9. Back of Tongue |
| 2. Teeth | 6. Uvula | 10. Pharynx |
| 3. Teeth-ridge | 7. Blade of ton. | 11. Epiglottis |
| 4. Hard plate | 8. Front of ton. | 12. Vocal cords |
| | | 13. Tip of tongue |
| | | 14. Glottis |

tract, and vocal cords (see Fig 1.1): The air necessary to produce speech is supplied by the lungs. The waveform shape is generated by the vocal cords with the vocal and nasal tract acting as filters.

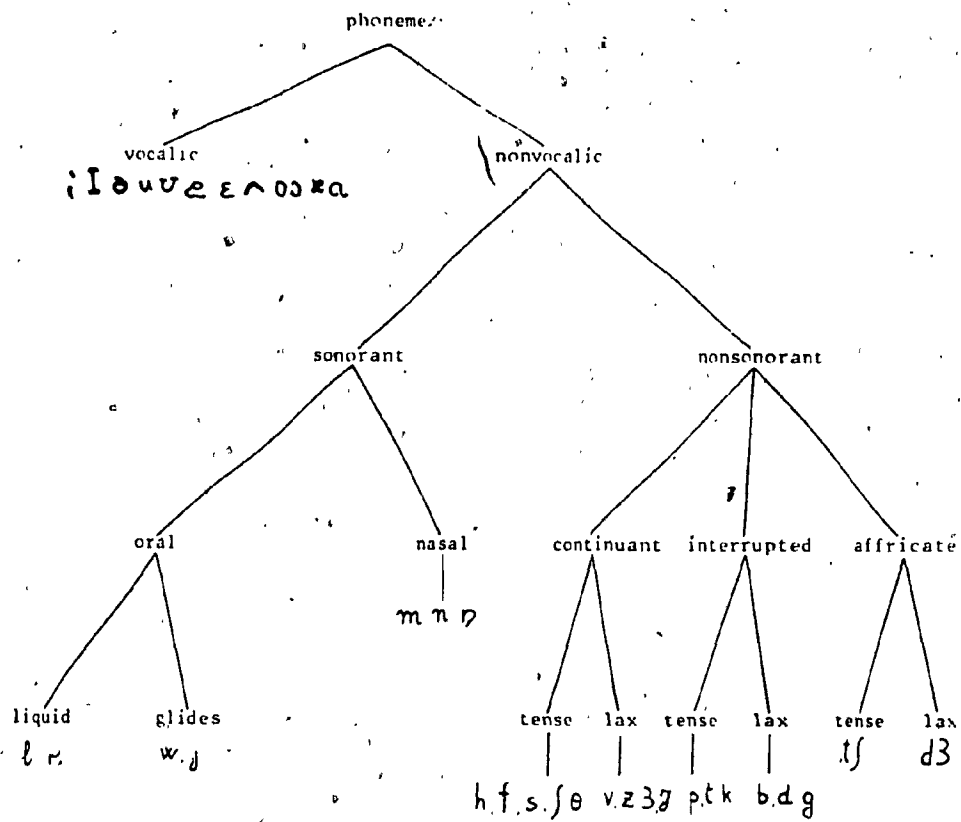
Speech sound thus produced can be separated into two different classes: Voiced and Unvoiced. Voiced sound is produced by a vibratory motion of the vocal cords. The natural frequencies of the vocal cords are called formants. They appear as f_1 , f_2 , f_3 , and so on. Only the first 3 formants are considered in speech analysis. Voiced sounds have typical fundamental frequencies: 120 Hz for men, 220 Hz for women, and 300 Hz for children [1].

Unvoiced or voiceless sounds are produced by a turbulent flow of air caused by an extreme narrowing of the vocal cord at some point. The frequency ranges of these sounds are rather wide and flat and vary from 4000 Hz to 8000 Hz.³

1.2 Linguistic Aspect Of Speech

The waveform generated by the vocal cords is converted into linguistic form at the listener's end. In turn, the listener fits his auditory sensations into sequences of words and sentences. The words are merely representations of acoustic symbols. However, words are formed by smaller linguistic units called the phonemes. Phonemes on their own do not convey any concept unless they are put together

Fig 1.2 Tree Representation of Phonemes of American English



with other phonemes. These linguistic units are strictly language dependent. Fig. 1.2 shows a tree representation of phonetic features and a phonetic representation of the standard English language [28].

Phonemes combined into larger units are called syllables. Syllables on their own or put together with more syllables form words. Therefore, words could be mono-syllabic or multi-syllabic. Different rules dictate the possible combinations of phonemes in forming syllables. Syllables usually have a vowel as the central phoneme surrounded by one or many consonants. The phoneme may be spoken differently at different times depending on the context representing the same basic unit. The different realizations of the same phonemes are called "allophones".

1.3 Acoustic Characteristics of Phonemes.

Speech sound is divided into two main groups, Vowels and Consonants. The basic distinguishing features of different phoneme classes are described here.

Vowels.

Vowels are produced through an open vocal tract, thus making a voiced source. The key characteristic of vowels is that they possess substantial energy in the low and mid-formants [3]. Different vowels can be distinguished by the position of the tongue, jaw, and the rounding of the

lips. The formant frequencies can be determined by the shape of the vocal tract. The second formant rises when the tongue moves forward.

There are three different categories of vowels. The "front vowels" (e.g./i,I,e/), the "back vowels" (e.g./u,U,o,a/), and the "central vowels" (e.g./aI,æ, /). Vowel recognition can be carried out by locating and measuring the first three formants. Variations in the length of the vocal tract cause the vowels of different speakers to be different [4].









Certain types of consonants such as labial, dental, and velar have their own significant effects on the formant transitions to or from an adjacent vowel. The effect shifts the formants in the middle of the vowel, thus making it hard to classify the vowels. Another problem arises due to "lax" and "tense" vowels in the English language. The same vowel in its "lax" and "tense" form has a duration difference. This duration difference is caused by vowel stressing. Also, vowels followed by voiced consonants are usually longer than those followed by unvoiced consonants [4]. Fig 1.3 shows the manner-of-articulation of the vocalic sounds with examples.[30]

Consonants.

Consonants are divided into several groups according to the "manner-of-articulation". The five major groups of

Fig 1.3 Generation of Vocalic Sounds

Tongue hump position.

	front	central	back
high	i (EVE) I (IT) 	ɜ (BIRD) 	u (BOOT) U (FOOT) 
medium	e (MATE) ɛ (MET) 	ʌ (UP) 	o (SEY) ɔ (ALL) 
low	ɒ (AT) 		ɑ (FATHER) 

consonants in the English language are, plosives, fricatives, nasals, glides, and affricates. Depending on the "manner-of-articulation", these groups share different acoustic properties. The sharp rising of the first formant is a fair enough indication that the sentence started with a consonant. The place of articulation can be found using the transitions of the second and third formants. Labial consonants have an overall lower formant. The second and third formants have a tendency to rise for dental consonants and for velar consonants these formants are closer to each other. A brief study of the various consonant groups is described below.

i) Plosive (Stop) Consonants .







The plosive consonants (Stop Consonants) (B,D,G) are voiced and (P,T,K) are voiceless. These consonants begin with a brief silence followed by an abrupt increase in amplitude at the releasing point of the consonant. Hence, the amplitude of the burst will be significantly different between the voiced and unvoiced consonants [5]. The distinguishable features of plosives depend on the adjacent phonemes. Fig 1.4 shows the place of articulation of the stop consonants.

Stop consonants may occur at the beginning of a word, within the word, or at the end of the word. Stop consonants which occur within a word could easily be confused for the end of the word during speech recognition.

eg: /B/ be
/D/ day
/G/ go
/P/ pay
/T/ to
/K/ key

Fig 1.4 Place of Articulation of Stop Consonants

Place of articulation

	labial	alveolar	palatal
voiced	<p>b (be)</p> 	<p>d (day)</p> 	<p>g (go)</p> 
unvoiced	<p>p (pay)</p> 	<p>t (to)</p> 	<p>k (key)</p> 










ii) Fricatives.

The fricative sounds are produced by a turbulent air flow. There are voiced fricatives (/V/, /DH/, /Z/, /ZH/), and unvoiced fricatives (/F/, /TH/, /S/, /SH/, /H/). High zero crossing rate and a small increase in high-frequency energy when total energy is low are some of the clues to detect weak fricatives. Voiced fricatives have greater low-frequency energy at the beginning of the fricative sound. Unvoiced fricatives are longer than voiced fricatives. Refer to Fig 1.5 which shows an illustration of how these sounds are produced.

eg:	/V/	vote
	/DH/	then
	/Z/	zoo
	/ZH/	azure
	/F/	for
	/TH/	thin
	/S/	see
	/SH/	she
	/H/	he

Fig 1.5 Generation of Fricative Sounds

Place of articulation:

	labio-dental	dental	alveolar	palatal	glottal
voiced	<p>v (vɒl)</p> 	<p>ð (ðɪθ)</p> 	<p>z (zɒz)</p> 	<p>ʒ (ʒʒɛ)</p> 	
unvoiced	<p>f (fɛ)</p> 	<p>θ (θɪθ)</p> 	<p>s (sɪt)</p> 	<p>ʃ (ʃɛ)</p> 	<p>h (hɛ)</p> 

iii) Nasals.

In the English language, the nasal sounds (/m,n/) are usually adjacent to a vowel. Because of the presence of the nasal, the adjacent vowels are also subject to nasalization. A significant increase in the bandwidth of the first formant is an after-effect of nasalization. The first formant is usually stationary when these sounds are produced but the second and third formant vary considerably from speaker to speaker[8].








eg: /M/ me
 /N/ no

Glides.

Glides (/w,r,l,y/) and nasals are similar in that they both occur exclusively beside vowels. Glides and nasals together are also known as "sonorants". Each glide sound has a unique formant characteristic [5]. The formant transition into a vowel is very smooth and slow and the dynamics of the transitions to adjacent vowels are the distinguishing characteristics of these sounds. Refer to Fig 1.6 for an illustration and classification of Glides and Nasal sounds.

eg: /Y/ you
 /W/ we
 /R/ read
 /L/ let

Fig 1.6 Classification of Nasals and Glides

	labial	alveolar	palatal
	m (we) 	n (no) 	ŋ (sing) 
glides	w (we) 		j (you) 
semivowels		l (let) 	r (rare) 

iv) Affricates and Diphthongs.

Combinations of vowels are called diphthongs, whereas, combinations of consonants are called affricates. The sounds, (/EY, IH/, /IH, UW/, /OY, IH/, /AW, UH/, /AY, IH/, /OW, UW/) are the diphthongs and the sounds, (/CH, SH/, /JH, ZH/) are the affricates in the English language.

Diphthongs and affricates have the same formant transitions as glides. The transitions take place to one vowel position followed by a short steady state and then motion to a second target which has a longer duration [5].

eg:	/EY, IH/	say
	/IH, UW/	new
	/OY, IH/	boy
	/AW, UH/	out
	/AY, IH/	I
	/OW, UW/	go
	/CH, SH/	chew
	/JH, ZH/	jar

The characteristics of each class of phonemes described above are not independent. When speech is produced by human beings, a prototypical set of features is not generated. The listener somehow aligns the features properly, discards the bad ones, and recognizes the speech correctly.

Phonetic detection algorithms are usually formalized in

such a way as to accept a large set of features jointly, rather than looking for one particular feature. The speech recognition systems based on acoustic-phonetic-recognition (APR) are called feature-based systems.

1.4 Speech Recognition By Computers.

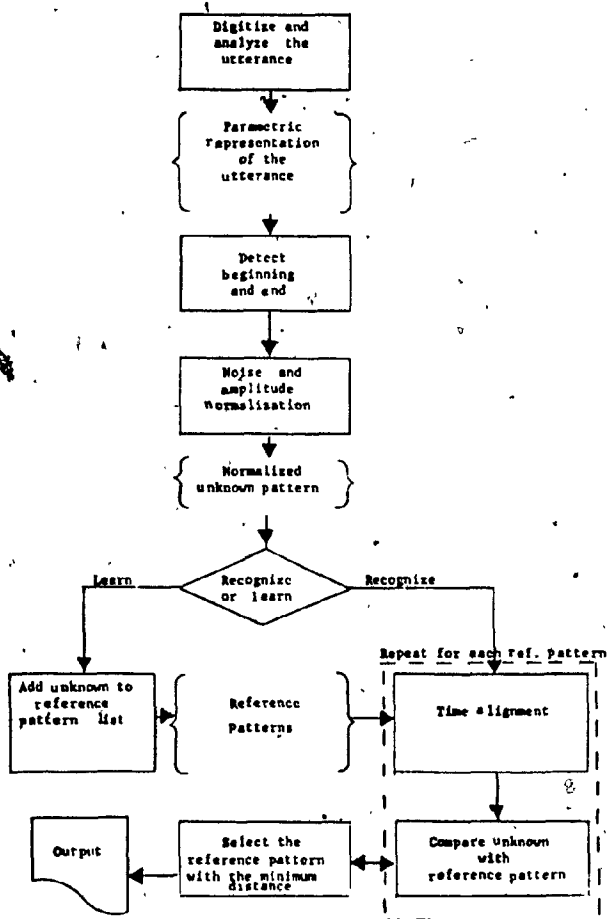
Speech recognition by computers started in the early fifties and has taken various approaches since then. The two types of speech decoding models which are currently used are the "active" model and the "passive" model [29]. The passive model is, for the most part, used in isolated word recognition systems. The basic necessary steps involved in this model are acoustic processing, feature extraction, and pattern recognition. Fig 1.7 shows a typical isolated word recognition system based on this model.

The active model is used in more complex task decomposition problems such as continuous speech recognition systems and speech understanding systems. This model is based on Knowledge Sources at various levels of signal interpretation. Fig 1 .8 shows a model of such a system.

Speech decoding research is currently being done in the following major areas:

- a) Isolated word, Speaker-dependent systems for limited vocabularies.
- b) Isolated word, Speaker-independent systems.
- c) Continuous Speech Recognition Systems.

Fig 1.7 A Passive Model Speech Recognition System



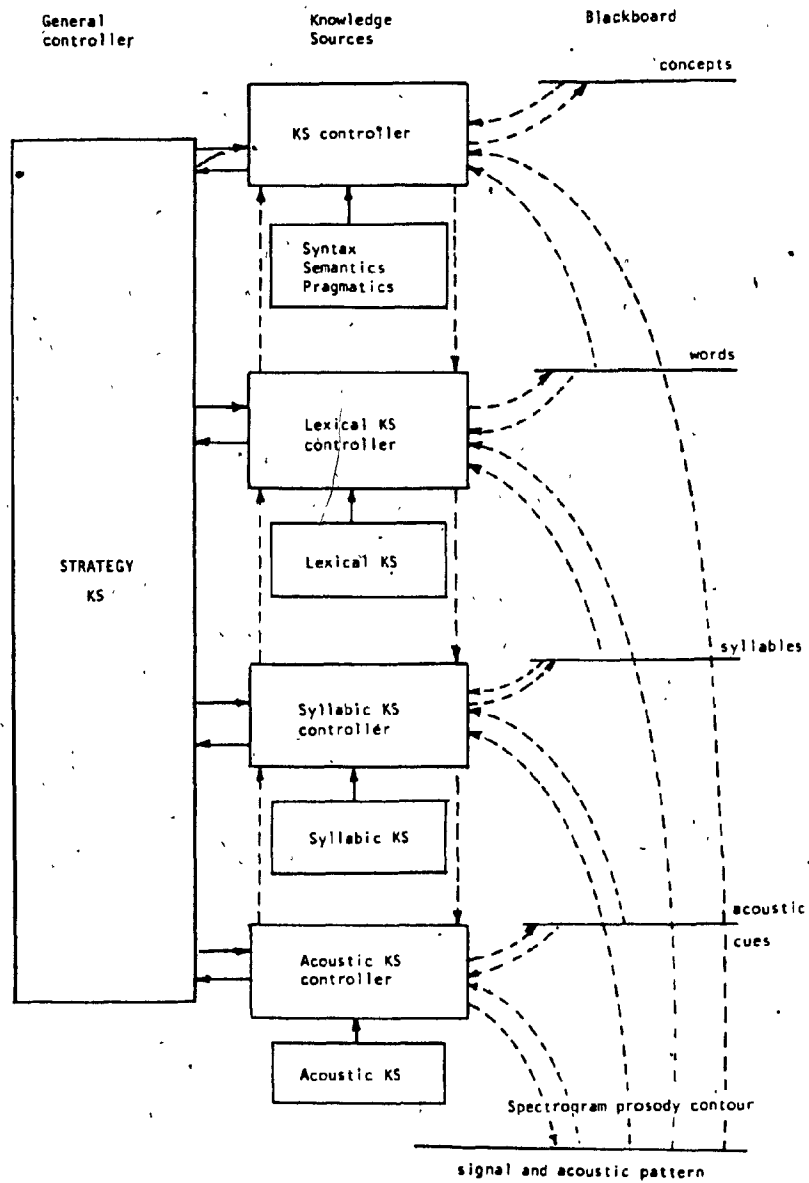
d) Speech Understanding systems.

Speech recognition involves a variety of principles which include challenging tasks such as detecting vowels and consonants, matching pronunciation of words, and making use of the prosodic, syntactic, and semantic features. Modern speech recognition systems employ either some or all of the above constraints. Appendix A shows some of the major institutions which are active in the Speech Processing field.

After more than 35 years of research, automatic recognition of natural or conversational speech is still beyond reach. Advanced developments in digital technology have substantially contributed to the recent advances in speech recognition systems.

Why is speech recognition so difficult? The main reason is that the acoustic signal is extremely variable. This variation comes from the speaker's dialect, his personality, and his emotional conditions. The chances that a selected speaker will speak the same word twice and the resulting acoustic signal will be identical is very slim. Word boundaries may not be apparent in the signal. Silent gaps in the same word may be confused with word boundaries. Numerous such problems make it difficult for computers to recognize speech.

Fig 1.8 An Active Model Speech Recognition System
(after De Mori)



1.5 Speech Recognition Systems: Past and Present.

It was evident to early researchers that the speech signal could be decomposed into simple sinusoidal wave-forms by analysis [2]. These decompositions could be approximated by electronic "filters" which separate the different frequencies.

The complexity of speech recognition by machine increases as we seek for speaker-independency and continuous speech recognition under poor acoustic environments. By carefully avoiding these complexities, early researchers focused their attention on isolated word recognizers for a small vocabulary set for a selected group of people.

Sonographs of speech signals were used by Dreyfus-Graf in the '50s in order to recognize human speech. The first speech recognizer developed by Davis, Bidduph, and Balashek of Bell Telephone in 1952, used zero-crossing count of the speech signal filtered at 2 different frequencies. Later on, many more speech recognizers were developed using more filter banks.

A major break-through in isolated-word, speaker dependent speech recognition occurred when Dudley and Balashek [33] developed a system which would segment the word into phonetic units. Perfect recognition was achieved for a single speaker. During the 'sixties, more word

recognizers were available with greater expanding capabilities. Some word recognizers were able to handle a vocabulary size of 500 words. Another system developed by Gold could recognize 10 speakers with 86% accuracy. The system developed by Medress was based on extracting more linguistic information from the speech signal.

The first commercial speech recognizer was developed by Threshold Technology in 1972. Recognition difficulties increased as the number of words in the vocabulary increased and the need for speaker independency came into the picture.

Commercially available speech recognizers are effective only within narrow limits. Problems such as, small vocabulary, need for training, misrecognition, and high cost are some of the major drawbacks. Most of these recognizers use Linear Predictive Coding for feature extraction and Dynamic Programming algorithms for time normalization and template matching. Even though these techniques produce very good recognition rates, ordinary small-scale microprocessors are not capable of performing such tasks. If low cost systems are desirable then they must use simpler techniques. Nippen, Verbex, Bell Labs, and Threshold Technology are some of the leading manufacturers of commercial speech recognizers. Table 1.1 shows some of the current speech recognizers available in the market for commercial applications [10].

Table 1.1

Manufacturer	Model	Speaker- independent	Connected speech	Price in \$	Error Rate %
Verbex	1800	Yes	Yes	65,000	0.2
Nippon Elec.	DP-100	No	Yes	65,000	1.4
Threshold Tech	T-500	No	No	12,000	1.4
Interstate Elec	VRM	No	No	2400	2.9
Heuristics	7000	No	No	3300	5.9
Centigram	MIKE 4725	No	No	3500	7.1
Scott Insts.	VET/1	No	No	500	12.6

From the evaluation table, it can be concluded that, for a reasonably good recognition system, the cost is too high. Therefore, a low-cost recognition system with an acceptable recognition rate and a reasonably sized vocabulary (of size 50) has many applications.

Voice activated systems are used in air traffic control, computer-aided design, production and process control etc. Connected speech recognition systems can be useful for blind people to speak into a system, for dictating machines, and finally, the listening typewriters.

The purpose of this thesis work was to develop a low-cost, speaker-independent, isolated-word speech recognition system for a small vocabulary set. Chapter 2 explains the development of such a system in more detail. Due to hardware limitations, the goal was not fully achieved and the work was redirected towards a continuous speech recognition system which is under development on a Vax-780 machine. This work is explained in Chapters 3 and 4. The results and progress are reported in Chapter 5.

Chapter II

A SPEAKER-INDEPENDENT ISOLATED WORD RECOGNITION SYSTEM

In this chapter we discuss the implementation of an isolated-word recognition system.

The recognition system can be considered as a Passive Model. Multiple templates were created during the learning stage for each word in the vocabulary as reference patterns. These multiple templates were created using the clustering algorithm. Many aspects of speech recognition systems, such as, end-point detection, time alignment, feature extraction, and distance measure were considered in developing this system.

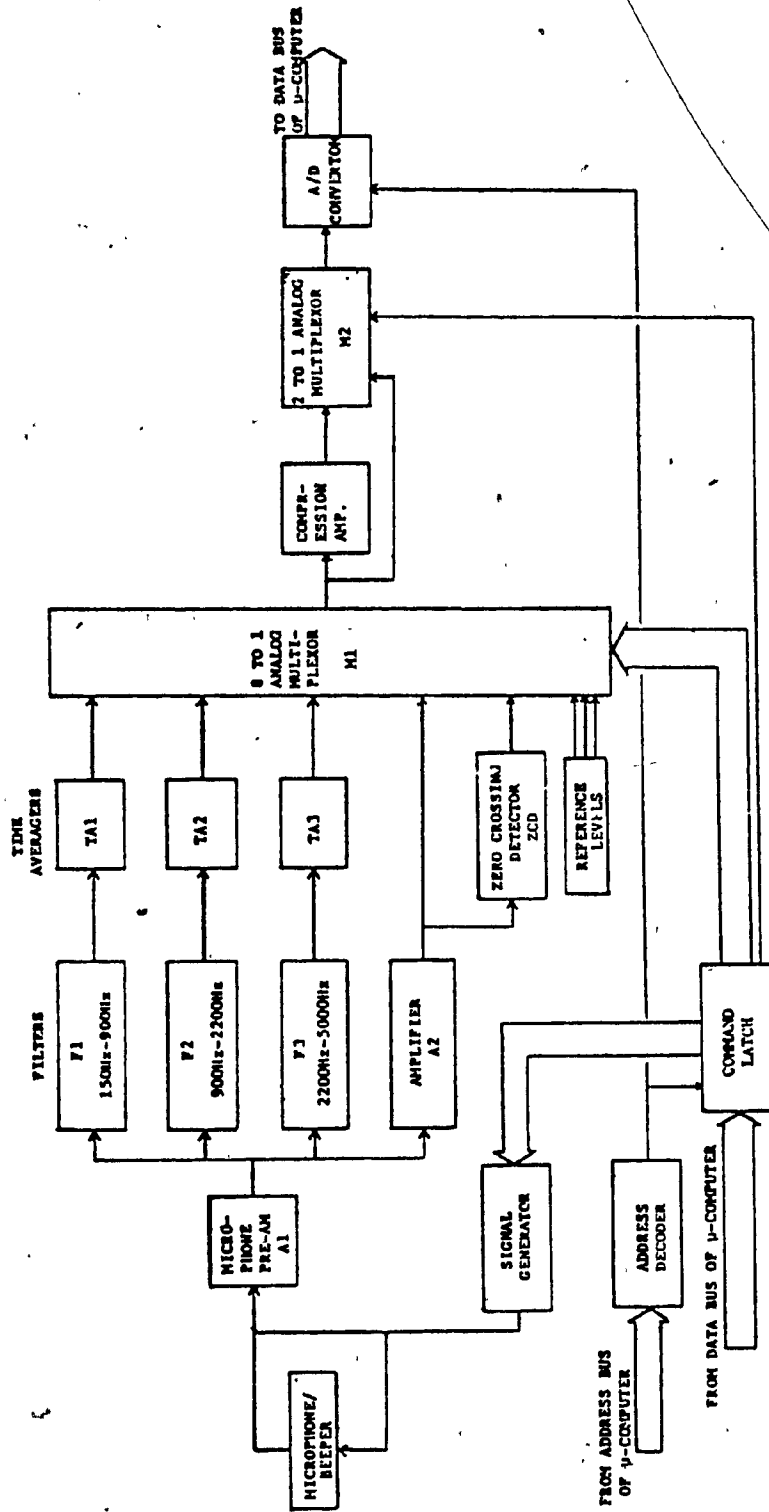
2.1 Hardware Organization Of The System

This system was developed on a low cost microprocessor system which consists of a Z80 cpu, 48K RAM, and floppy disk drives. The speech board was made by Heuristic Inc. (Los Altos, California).

The speech board consists of three audio bandpass filters, one overall zero-crossing detector, a compression amplifier, A/D converter, and an analog multiplexer. Fig 2.1 shows the hardware configuration of the speech board.

The incoming speech signal is filtered in three major

Fig 2.1 Schematic Diagram of the Speech Board



frequency domains. The ranges of the filters are, 150Hz to 900Hz (f_1), 900Hz to 2200Hz (f_2), and 2200Hz to 5000Hz (f_3). Each filter values corresponds to the approximate frequencies of the formants f_1 , f_2 , and f_3 of the average vocal tract.

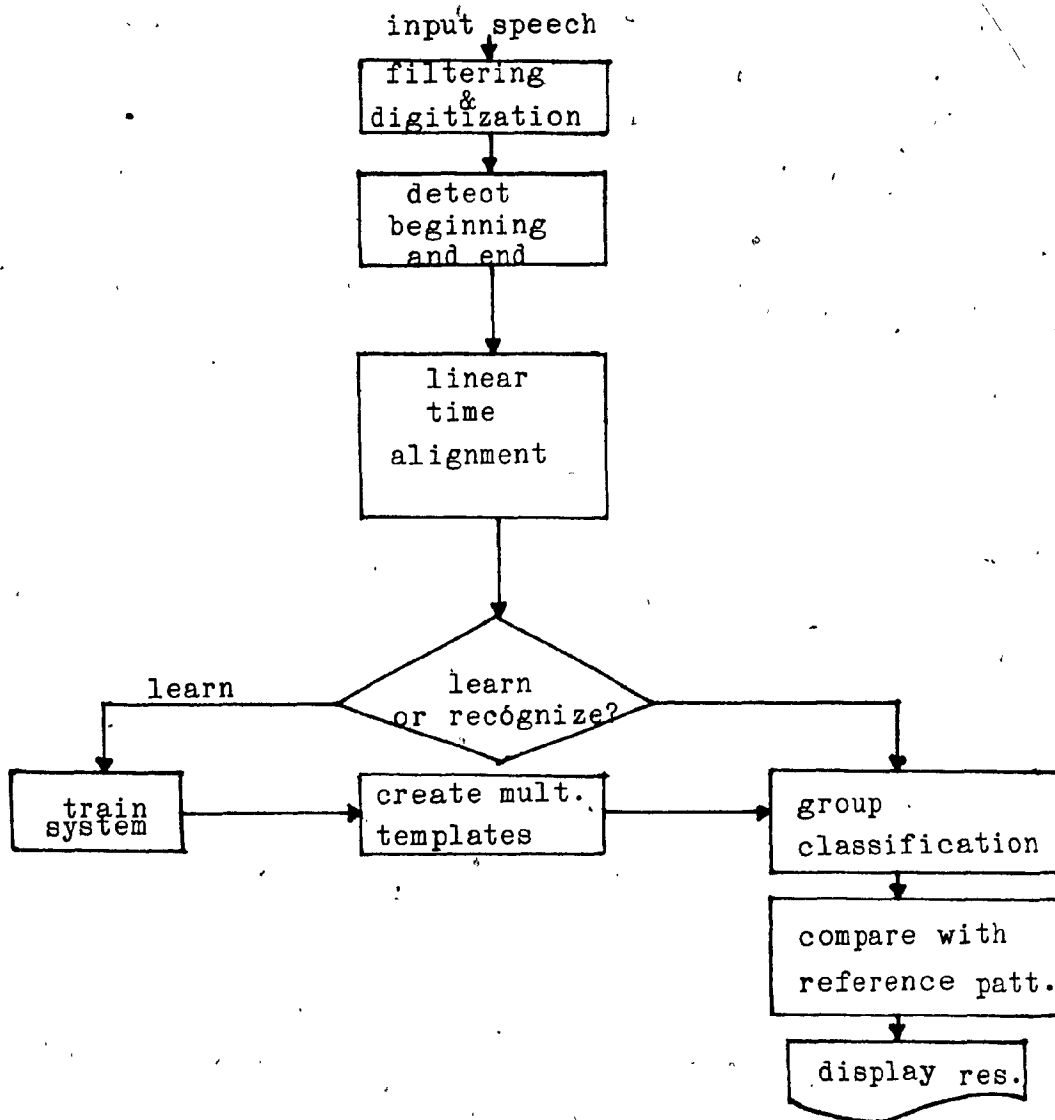
The speech signal is sampled at every 10 msec for 1.5 sec. Thus, 150 samples are collected from each band amounting to a total of (150×4) 600 data samples per utterance. Fig 2.2 shows the algorithm for preprocessing and recognition which constitute the software part of the system.

2.2 Preprocessing.

A beep from the microphone marks the beginning of the speech window and the speaker begins to speak. After 1.5 sec another beep marks the end of speech collection. The speaker may be slow in his response and may start speaking moments later. Therefore, the speech samples may have sets of data which do not contain any relevant information. The speech is also accepted in a noisy environment but actual speech should not be confused with noise. Therefore, the beginning and end time of the actual length of the speech must be detected before any further processing.

The algorithm used here is similar to the one proposed by Rabiner and Sambur [25]. The idea is that the measured samples from band 0, 1, and 2 are added together and checked.

Fig 2.2 Preprocessing and Recognition



against a threshold value starting from time, $t=1$.

This threshold value is lowered, or the beginning point or the end point of the speech is moved backward or forward, if the zero-crossing count at that point is greater than a (another) threshold value. This is because the fricative sounds have low energy but higher zero-crossing densities. If the actual speech does not constitute at least 100 msec (minimum word length), then the system considers the utterance as poor data. The data will be rejected if either the beginning or end point cannot be detected. In order not to confuse a stop consonant appearing in the middle of a word as word-end, the minimum word length is checked. The stop consonant leaves a short silence after the consonant.

Selecting Evenly Spaced Samples and Time Wrapping

The actual speech data obtained after performing the begin-end algorithm is stored in a temporary buffer. The length of the speech may vary even for the same word spoken at different times. Therefore, a standard length has to be maintained for every word spoken. For this reason, a set of evenly spaced samples are selected from the speech data. The selection must be made in such a way as not to lose any vital information on the speech.

The standard length is kept as 64 samples/word. It is obvious that this represents only the first 160 msec of the utterance thus losing a considerable amount of information.

There are various techniques used by researchers for retaining a small set of data from a large set with a maximum amount of information.

Linear Interpolation with translation is used here since it has reported very good results. Translation is used for time warping. Time warping eliminates timing difference between two utterances. Linear time alignment means that two utterances to be compared are stretched or compressed linearly so that they will have the same length. Non-linear time warping techniques such as Dynamic Programming were used in various isolated word recognition systems. However, it was shown that for monosyllabic words, linear time warping gives good results [27].

2.3 Test Data Collection

Speaker independent speech recognition systems must be capable of extracting speaker-independent features from speech pattern. Another way to achieve speaker independency is by storing multiple templates which will accommodate many variations of the same word. This is also known as speaker normalization. The number of such templates depends on the system capabilities such as memory availability.

The recognition system under consideration uses multiple templates for a 20 word vocabulary. The training samples are collected from a wide range of speakers. Native and non-native English speakers (male and female) were included as candidates for data collection. 32 speakers (8

native English speakers, 8 female native English speakers, 8 non-native male English speakers, and 8 non-native female English speakers) were asked to come at 4 different times and each time 4 samples of each word were collected giving $(32*4*4)$ 512 speech patterns/word.

The vocabulary set consists of 10 command words, STOP, START, ERASE, ENTER, RUBOUT, GO, NO, HELP, YES, REPEAT, and the digits from 0 to 9. Considering the need for real time recognition and availability of RAM area to store the templates, 10 templates/word with a total of 200 were made for the 20 word vocabulary. The clustering technique was used to create templates from the large set of speech patterns.

Organization of Multiple Templates .

The objective is to create a small set of templates out of a large set of sample patterns so that the templates will represent the maximum variations present in the sample set.

Since variations between speakers are very large, the training patterns may span a large area in the feature space. Simple averaging of the training patterns is meaningless. Therefore, it is necessary to use some meaningful approach to extract templates.

There are various clustering methods available and the "Maximin" (maximum-minimum-distance) clustering algorithm [26] is used here. We consider each sample pattern which

Fig 2.3 Clustering Algorithm

- STEP 1: Find an arbitrary initial point in the data space. Let this be the 1st cluster point, C_1 .
- STEP 2: The point with the maximum distance from C_1 would be the second cluster point, C_2 .
- STEP 3: The successive cluster points, C_3 , C_4 , ..., C_n ($n=10$) are calculated by finding all the minimum distances to every existing cluster point and finding the maxima of all minims.
- STEP 4: If desired number of clusters are not found, goto step 3 else continue with step 5.
- STEP 5: Compute the domain for each cluster point. The domain is computed by finding all the distances between data points and cluster points. The points with minimum distances belong to the closest cluster point.
- STEP 6: Find the cluster center by calculating the average of all the points belonging to the same domain.

contains 64 speech samples as a data point in the space. We have 512 such data points for each word. These data points are classified into 10 domains and the midpoint of each domain is the desired template. The Maximin clustering algorithm is explained in Fig 2.3.

The cluster center found in Step 6 is the desired template. The algorithm is performed for every word in the vocabulary.

The number of templates (domains) for each word could be increased or decreased. The larger the number of templates, the greater the diversity of speech patterns.

2.4 Recognition Procedure

Recognition using the template matching technique is basically applied here. The first three formants and the zero-crossing count are the features used for matching. In the usual way of template matching, every template is matched against the target word (unknown word) and the word which gives the minimum difference is considered as recognized. In our system we have 10 templates for each word and a total of 200 templates for the entire vocabulary. Matching against each word not only slows down the system but also affects the performance of the system. One solution to this problem is to classify the speech signal and the vocabulary set so that only one class (group) of templates is needed to be considered for matching.

Classification is done based on the acoustic features of the words. The entire vocabulary is classified into seven groups with respect to the acoustic behaviour. These features are speaker-independent and applied with "a priori" knowledge. The target word goes through classification first and receives a class-label.

Classification Technique .

Classification of the target speech pattern prior to template matching would give a better recognition rate and recognition time. Classification is done by making use of the "peaks" and "dips" on the zero crossing signal. Peaks may appear as, high-peak, medium-peak, or low-peak. The terms high, medium, and low correspond to the amplitude of the peak.

The peak-dip detection algorithm counts the peaks on the zero crossing signal and the beginning time (t_b), and the ending time (t_e) of the peaks. The peak-dip characteristic on the signal represents the acoustic property of the speech itself. The classification algorithm finds an overall distinguishable characteristic of the signals rather than locating or detecting any phonetic features.

The 20 word vocabulary is classified into 7 groups with class-labels as follows:

HSPB	High-Single-Peak (Beginning)
MSPB	Medium-Single-Peak (Beginning)
MSPE	Medium-Single-Peak (End)
HSPE	High-Single-Peak (End)
HMP	High-Multiple-Peaks
MMP	Medium-Multiple-Peaks
LMP	Low-Multiple-Peaks

Fig 2.5 illustrates the Peak Labelling Algorithm.

Fig 2.5 Peak Labelling Algorithm

```

repeat
  repeat
    find-peak( $p_n$ ,  $t_b$ ,  $t_e$ )
    case  $p_n$ 
       $\geq t_h$  :  $h_p := \text{true}$  (* high-peak *)
       $\geq t_m < t_h$  :  $m_p := \text{true}$  (*medium peak*)
       $\geq t_l < t_m$  :  $l_p := \text{true}$  (* low-peak *)
    end (* end-case *)
  until (  $h_p$  OR  $m_p$  OR  $l_p$  )
until (  $n = N$  )

```

where,

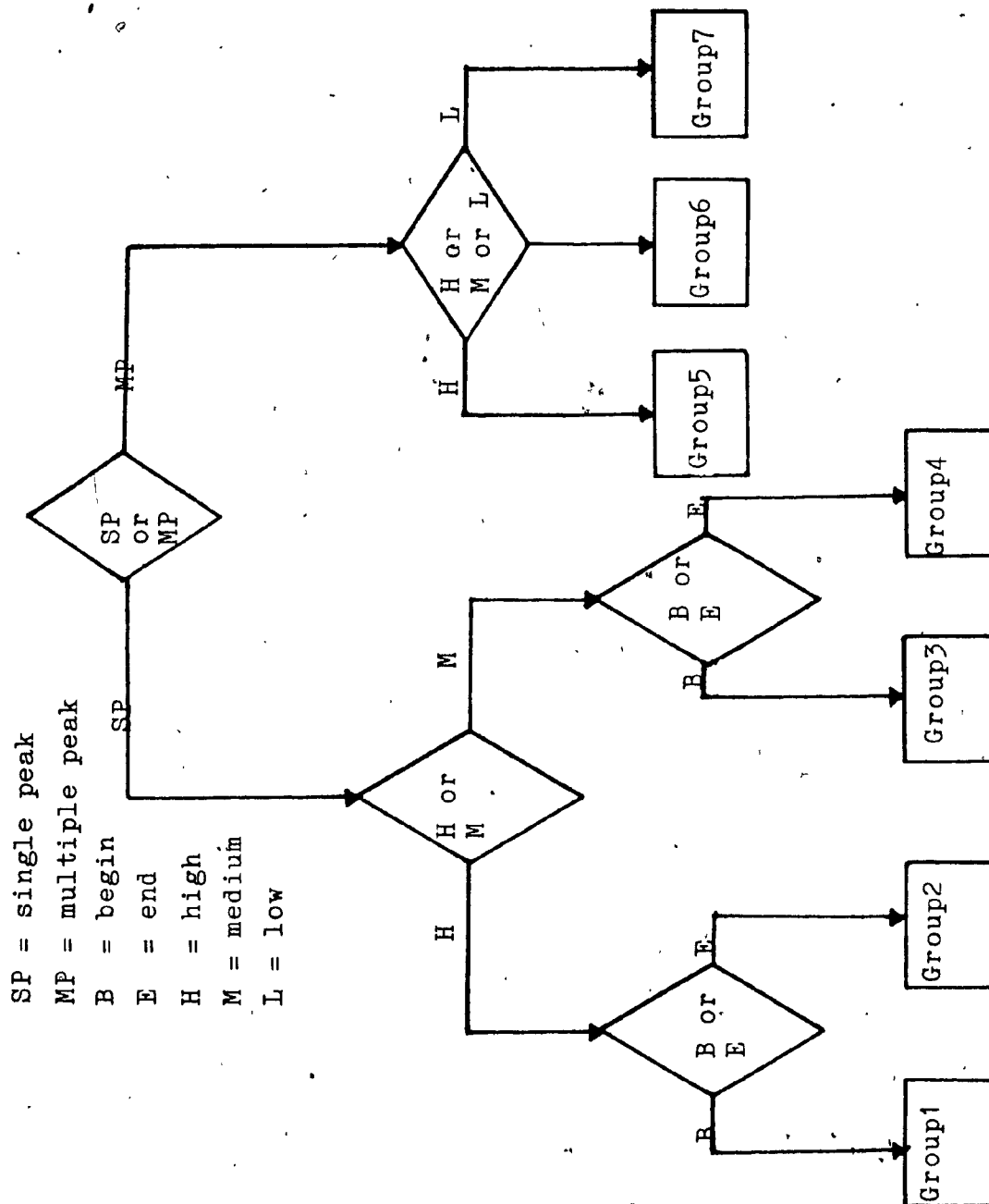
N = number of speech samples (64)

$t_{h,m,l}$ are thresholds

p_n = number of peaks

$t_{b,e}$ time beginning and time end.

Fig 2.6 Classification Algorithm



Once the number of peaks and the type of peaks are detected, the classification is carried out by the algorithm shown in Fig 2.6.

Each group described in the above algorithm contains the following words:

Group 1: SEVEN, START, STOP

Group 2: YES, ERASE

Group 3: HELP, FOUR, GO, TWO, ZERO

Group 4: REPEAT, ENTER, EIGHT

Group 5: SIX

Group 6: NINE, ONE, THREE, FIVE, (NO)

Group 7: RUBOUT, (NO)

The unvoiced fricatives sound "s" give rise to high zero crossing in Group 1, Group 2, and Group 5. In Group 2 this property (the high peak) will appear at the end of the speech and in Group 1 the peak will appear at the beginning of the speech. In Group 5, the "s" and "x" sounds of the SIX produce a multiple high peak at the beginning and at the end of the speech.

Group 3 contains words starting with fricative sounds and plosive sounds. These fricative sounds and plosive sounds have lower zero crossing rates and will appear at the beginning of the word. Group 4 contains the words with plosive sounds "p", "t", and "g" which will generate peaks towards the end of the speech.

Group 6 and Group 7 have words starting with sonorant sounds which have low zero-crossing counts and multiple peaks.

Template Matching and Recognition

The target word is subjected to go through the above described classification process before the actual recognition process (template matching) is completed. Unknown words are matched only against those words in the group into which they were classified.

A Weight Function is used in calculating the distance measure between the unknown and the prototypes. The weighted distance is calculated separately for each feature vector (f_0 , f_1 , f_2 , and z_0). The distance is calculated as follows:

$$W_0 = \sum_{i=1}^k |(f_{0i} - f'_{0i})|$$

$$W_1 = \sum_{i=1}^k |(f_{1i} - f'_{1i})|$$

$$W_2 = \sum_{i=1}^k |(f_{2i} - f'_{2i})|$$

$$W_z = \sum_{i=1}^k |(z_{ci} - z'_{ci})|$$

$$D = 1/W_0 + 1/W_1 + 1/W_2 + 1/W_c$$

where,

f_0, f_1, f_2, z_c

are the feature vectors of the templates,

f'_0, f'_1, f'_2, z'

are the feature vectors of the target word,

$1/W$ is the weight function

(W = the co-variance factor),

D is the weighted distance between the two patterns.

Distance is now calculated between every word in the same group and the target word and is stored in a Distance Buffer (DB). The word which contains the minimum distance in DB could be selected as the word recognized. However, instead of finding just the first minimum, the first 5 minimas are found and, in order to be a potential candidate for recognition, at least 3 out of 5 minimas must be present in the same word.

2.5 Results and Conclusion

The Real-time response of the system was excellent. The recognition rate was approximately 70% which is not within the acceptable range. However, various interesting observations were made :

- i) Multisyllabic words were easier to recognize than monosyllabic words since the features were

correctly detected.

- ii) The clustering technique is definitely a promising approach for creating multiple templates in order to achieve speaker-independency for a small vocabulary set.
- iii) The system could be easily implemented on a more powerful machine (e.g. a 16 bit microprocessor) which would help to extract more features for group classification.
- iv) Words classified into the correct group were always correctly recognized. Wrong classification was the major cause for misrecognition.
- v) The templates were made permanent and there is no need for further training the system for a new user.
- vi) One overall zero-crossing detector is not enough and there must be at least one zero-crossing detector for each band. This would give better distinguishable features between the speech signals.
- vii) Selection of words in the vocabulary set is also important. Since small set word recognizers have many applications, these words should be selected carefully so that they may be recognized easily.

study on continuous speech recognition systems and a contribution to the problem of accessing a very large lexicon.

Chapter III

CONTINUOUS SPEECH RECOGNITION

In this chapter we look at the problem of continuous speech recognition, the techniques used by researchers, and also the system under development at Concordia University.

By continuous speech recognition, we mean the capability of recognizing natural speech which contains long strings of words spoken with or without pauses between each word. The vocabulary size of such systems will depend on the particular language we are dealing with and the size will usually be more than 15000 words. Usual speech recognition techniques such as those used in isolated-word recognition systems are not adequate in this case because of the very large vocabulary size. Therefore, detailed study of acoustic signals, extraction of more features, and hypothesization of syllables becomes necessary to carry out such tasks.

Active model speech recognition systems must be considered in order to handle complex problems associated with continuous speech recognition. Rule-based systems for correctly predicting acoustic properties of speech sounds are also used by recent researchers [12]. Rule-based systems give more insight into the characteristic acoustic

features of speech sounds based on small data.

Segmentation is an important aspect of continuous speech recognition. Segmentation means, segmenting utterances into phonemes. This process may not be necessary in isolated-word recognition systems. Usually, the adjacent phonetic events will interfere with the phoneme under consideration thus making segmentation difficult. In situations such as "I scream" and "Ice cream", even the boundary detection of words will be difficult. Therefore, it is necessary to provide higher level knowledge. In traditional continuous speech recognition systems, a lattice of segment/label is produced first.

3.1 Continuous Speech Recognition Systems (CSR).

Several continuous speech recognition and speech understanding systems have been developed by incorporating various techniques. Appendix-A shows various institutions which are currently involved in different aspects of speech processing and the technique they are using.

The Hearsay I and II, the Harpy, the Dragon systems by Carnegie-Mellon, the Hwim (hear what I mean) system by BBN, the SDC systems, the IBM system, and the Lincoln system developed at MIT are some of the continuous speech recognition and speech understanding systems developed in the past. A brief look at the design criteria used in these systems are considered now.

Preprocessing, Knowledge representation, and Matching are the main design principles behind all the CSR systems. Preprocessing consists of three important concepts, namely, parametric representation of speech signal, labelling, and segmentation.

Dragon uses the amplitude and zero-crossing from a 5-channel-octave-filter as its parameters for the speech signal. Using these parameters, the system then computes the probabilities for each of 33 possible phonemic symbol for every 10 ms segments of speech. In order to accommodate allophonic variations, the system uses multiple reference templates for each phonemic symbol.

Hearsay-I also uses 5-channel-octave-filter to represent its parameters for speech in the form of amplitudes and zero-crossings. Every 10 ms of speech is then classified and given a phonemic label based on a predefined set of cluster centers. Syllable-like segments are then obtained from each cluster with the same label.

Unlike Dragon and Hearsay-I, the Lincoln system uses the LPC spectrum analysis and tracks formants to use as parameters. It performs a preliminary segmentation and labels the segments as vowels, fricatives, or stop sounds. The labels are further classified into acoustic-phonetic elements by computing further spectral measurements such as formant frequencies, formant measurements, and formant amplitudes etc.

The IBM system uses energy spectra and spectral changes for segmentation and adopts the same technique as template matching in Dragon for labelling.

Knowledge representation is the next major step involved. Knowledge may be represented as networks or as production rules. The CSR systems are normally furnished with three different knowledge sources: the phonological rules, lexicon, and syntax [31].

The Dragon and IBM systems have all the knowledge represented as a finite-state network which represents a hierarchy of probabilistic functions. The Lincoln system uses a set of production rules to represent knowledge. Hearsay-I organizes its knowledge as independent but cooperating modules which allow easy modification when necessary. The representation of knowledge in each module is rather different. For example, production and prediction (hypothesis) rules are used to represent syntax knowledge, whereas, the lexicon is simply a representation of phonemic base forms. Even though IBM and Dragon use network representation, Dragon evaluates the likelihood of all possible paths while IBM evaluates only the most likely paths.

Matching is the final step involved. A stack containing a list of alternative word sequences arranged in descending order with respect to their likelihood is used by all systems except Dragon. Each of these words is matched

against the unmatched symbol string to estimate the likelihood of occurrence. A new list of acceptable word sequences is generated and this process is repeated for the entire utterance.

The performances of these systems were different as reported in [31]. The Hearsay-I and Dragon used 102 sentences and the Lincoln system used 275 while the IBM used 363 sentences for its test run. Hearsay-I had 31% sentence accuracy and 55% word accuracy for 4 speakers and 5 trials. Dragon achieved 49% sentence accuracy and 83% word accuracy for a single speaker. The Lincoln system has 49% sentence accuracy for 6 speakers and 1 trial. IBM, with 81% sentence accuracy and 97% word accuracy had the best performance for a single speaker.

A comparison among these systems is not simple since the experiment was carried out differently. However, the overall performance gave an insight into the still existing problems. Better segmentation, labelling, and improved matching techniques are still needed.

3.2 Expert Systems in Speech Recognition.

The area of computer science where the machine has to see images, listen and understand voices, is still in its early stages. These are some of the areas where artificial intelligence can be applied so that the machine may emulate intelligent behaviour. It is also in these areas where researchers come up with a new kind of approach using Expert

Systems .

The expert systems are developed on the basis of a collection of facts, rules of thumb, and methods of applying those rules and making inferences [13]. These programs differ from conventional computer programs because their tasks do not entail algorithmic solutions, rather they infer or deduce conclusions based on available knowledge by applying pertinent rules. Expert systems were used previously in various other fields such as medical diagnosis [14], crew scheduling for space shuttles, LANDSAT imagery [15] and so on.

The primary source of expert's knowledge must come from at least one human expert who has adequate knowledge and experience in the specific field.

Properties of Expert Systems.

- a) Structure of expert systems is modular. Thus, the knowledge about the problems, inference procedures, global data base, and information about current problems (input) are well separated.
- b) Modularization of inference procedures allows parallel processing capability. (will be explained in detail later on)
- c) Any module of the expert could be changed easily whenever it becomes necessary. For example, if new knowledge becomes available it could be added

to the knowledge base without interfering with other modules.

Knowledge in expert systems is usually represented in the format of production rules. Rule-based systems operate by applying rules, making inferences, and applying further rules, if necessary, depending on the inference or inferences made previously. Rule-based systems are favorable if the system under consideration has to be constructed based on previous experience. The rule interpreter uses control strategies based on top-down(goal-driven), bottom-up(data-driven), or both.

The Expert System Under Consideration

A speaker-independent continuous speech recognition system under development, as we have seen earlier, must be capable of performing complex tasks. The numerous processes which carry out such tasks may have to work directly on the speech pattern for extracting various features. By taking into account various objectives, the speech recognition system has the following design properties:

- a) Parallel execution capability. Since real-time recognition is desired, parallelism is a must.
- b) A distributed knowledge module for problem solving. Using this, it is possible to separately update each piece of knowledge when new knowledge becomes available.
- c) A control strategy capable of scheduling sensory

procedures will extract new cues from the data when it is necessary for confirming or supporting an inference currently made.

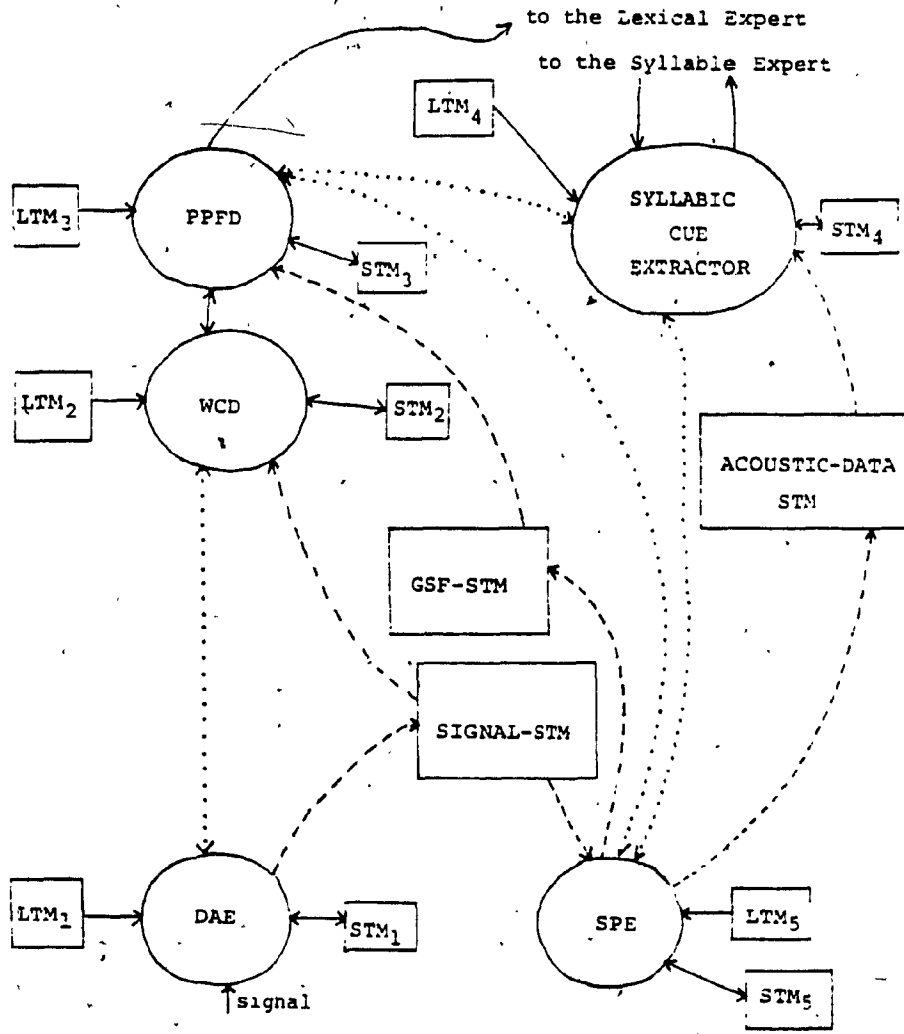
In consideration with the above design requirements, a system has been developed as proposed by De Mori.[16] which extracts acoustic cues, generates syllabic hypothesis, and accesses a very large lexicon. The cooperation of computational activities has been conceived using the paradigm of an Expert System Society [17].

Experts are computing agents which execute reasoning programs using structural and procedural knowledge in an integral form. They cooperate in extracting acoustic cues from the signal, in generating hypotheses about bounds of syllabic segments and phonetic feature hypothesis inside the segments.

Fig 3.1 shows the experts Exp_j ($1 \leq j \leq 5$) of the Auditory Society, their LTM, STM, and their communication links. Each expert is associated with a Long Term Memory (LTM) containing that Expert's knowledge and a Short Term Memory (STM) where data interpretations are written.

Each expert may create an instantiation of itself which applies its knowledge to a particular set of data in some particular context. An instantiation may communicate with other instantiations of the same expert or with other experts. The experts do not communicate through a common

Fig 3.1 Expert System Society for Speech Decoding



----- shared memory access lines
 inter-expert communication lines

data-base. They are provided with an elaborate control strategy.

3.3 Representation of Expert's Knowledge

The knowledge of the experts contains context-sensitive rules. It controls the extraction of spectral cues from speech data, produces a description of the extracted cues, and generates phonetic hypotheses. The algorithms for generating descriptions of acoustic data and for generating hypotheses of corresponding phonetic features are expressed in a frame language.

Frame language is particularly suitable for integrating structural and procedural knowledge, handling context-sensitive knowledge, and making inferences. A frame is an information structure comprising a frame-name and a number of slots. A slot contains information concerning a particular item called a slot filler [17]. Slot fillers may include descriptions of events, relations, results of procedures, and invocations of other frames. Fig 3.2 shows the physical structure of a frame and its components.

Slots can be filled by the results of sensory procedures invoked for extracting cues from data. The content of already filled slots could be used as a contextual reference for those sensory procedures not yet executed. A slot entry may be restricted by conditions where predicates have to be evaluated or where the evaluation of predicates may require further evaluation of

the functions which need semantic information. The slot filling structures are defined by a context-free grammar whose re-writing rules are shown in Table 3.1.

The terminal symbols are written in lower case letters and the non-terminals in upper case. The starting symbol is <FRAME>. The quantities appearing inside the brackets are optional. An exponent k is applied whenever the content of the base is repeated k times, in a sequence.

TABLE 3.1

Rules of the frame-structure grammar

<FRAME>	:=(<NAME> <SLOT-LIST>)
<SLOT-LIST>	:=(<NAME> [(<DESCRIPTION>)])k>0
<DESCRIPTION>	:=(<described-as <CHDES>)
	:=(<CONNECTIVE> <DESCRIPTION>k>1
	:=(<not <DESCRIPTION>)
	:=(<filled-by <FRAME>)
	:=<CONDITIONAL>
	:=(<result-of <PROC>)
<CONDITIONAL>	:=(<when <PREDICATE EXPRESSION>
	<DESCRIPTION>
	[(<else <DESCRIPTION>)]
	:=(<unless <DESCRIPTION><DESCRIPTION>)
	:=(<case <NAME> of
	(<DESCRIPTION> filled-by
	<FRAME>)k>1
<CONNECTIVE>	:=or
	:=and
	:=xor
	:=sequence

TABLE 3.1 (contd.)

<PREDICATE EXPRESSION>	:=<PREDICATE> :=(not (PREDICATE)) :=(<CONNECTIVE>(<PREDICATE>k>1)) [*]
<PROC>	:=F-<function> :=P-<procedure>
<NAME>	:=any string of characters
<CHDES>	:=any cue or hypothesis description

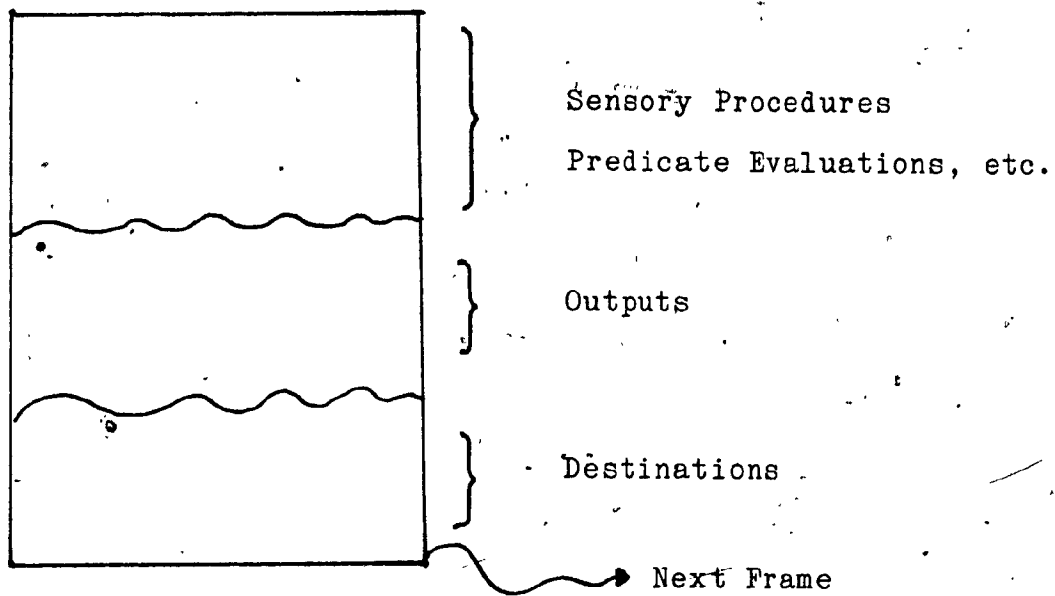
The knowledge stored in the LTM of an expert is a collection of algorithms, each having a frame structure. When the execution of an algorithm is invoked, an instantiation of the first frame of the corresponding structure is created into the expert's STM. The slots of the frame get filled while the expert executes the algorithm. Again, an attempt to fill a slot may cause instantiation of another frame and this operation can be done recursively.

When all the slots of a frame instantiation are filled, the frame instantiation is completed.

The slot described-as <CHDES> gets filled by generating descriptions of acoustic cues or interpreting hypotheses. The execution of a procedure can be initiated by trying to fill that particular slot of the frame. A procedure in a given instantiation has access to all the slots which have already been filled for that particular instantiation. The knowledge of the procedural rules must come from human experts who have previous experience in recognition systems.

The slot filled-by <CHDES> corresponds to the instantiation of a frame represented by its NAME. The slots with connective descriptions may cause the invocation of other frames and execution of procedures for extracting new cues if necessary for evidence. The connective sequence implies that time consistency must be maintained while

Fig 3.2 Frame Structure



describing the temporal sequence of events such that the $(i+1)^{\text{st}}$ event must begin at the end of the i^{th} one.

Heuristic knowledge is incorporated in order to keep the number of instantiations as low as possible when instantiations cannot reach completion. Conditionals and preconditionals are also used for this purpose. Default conditions are also used for filling frame slots. Frame instantiations that remain incomplete do not contribute any descriptions. If this happens, the frame will not be placed in the experts' STM.

The frame language structure allows one to express complex inferences and cue extraction rules very easily. Invocation of frames also corresponds to the sub-goal approach used in artificial intelligence. In this case, filling a slot is a sub-goal and the frame itself the main-goal.

3.4 Interaction of Experts to Decode Speech

Each circle in Fig 3.1 is an expert. Each expert has its own task to perform as described below.

The first expert is the Data Acquisition Expert (DAE). The speech signal is first sampled, quantized and stored into the "SIGNAL-STM". The DAE pre-emphasizes the signal and computes the gross spectral features (GSF) and stores it into the GSF-STM. It then looks for the beginning of the speech and starts sending messages to the next expert, the

Waveform Cue Descriptor (WCD). The knowledge needed for detecting beginning and end-points, windows for signal processing, and various other thresholds are stored in the LTM of the DAE.

The next expert, the Waveform Cue Descriptor, describes the total energy signal in terms of peaks and valleys with respect to time and based on zero-crossing densities. The DAE could still be functioning while the WCD is performing the description task. The alphabets of the descriptions, also called the Primary Acoustic Cues, are shown in Table 3.2.

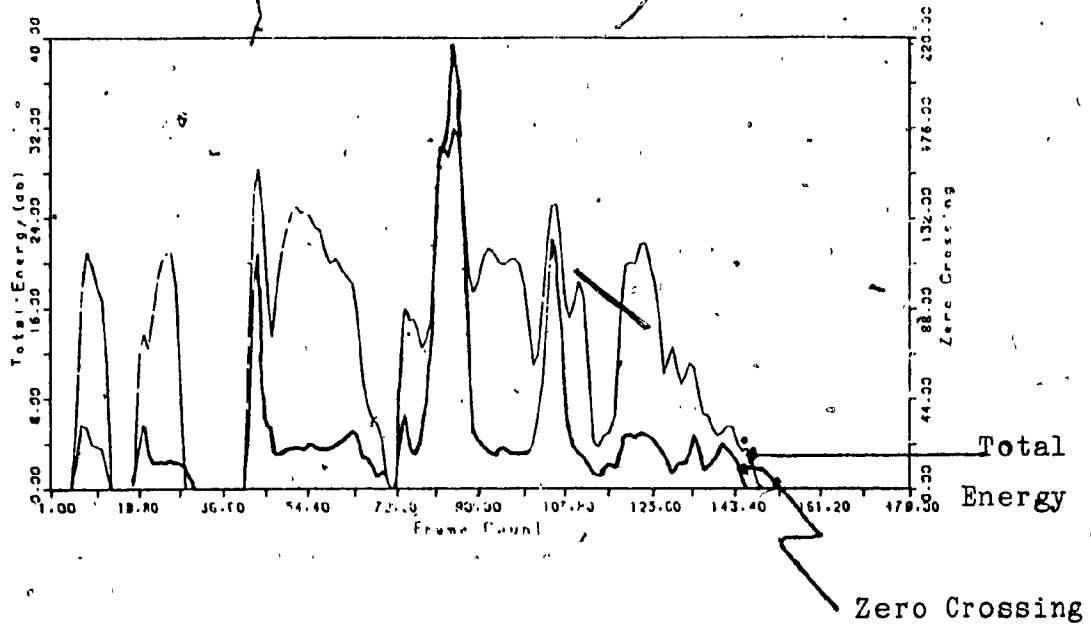
Short and Long refer to the duration of the peak or dip based on time while high, medium, and low are based on the amplitude.

With every detected symbol, the beginning time (t_b) the end-time (t_e), and the zero-crossing count are included.

The extraction of these cues does not require any contextual constraints. The algorithm for the extraction and description could be found in [12]. A set of such acoustic cues is generated by the WCD for the sentence, "A good turn deserves another", spoken by a male, native English speaker, is shown in Fig. 3.3. This particular sentence will be used later for illustration purposes.

Fig 3.3 Generating Acoustic Cues from Energy Signal

"A Good Turn Deserves Another"



```

L-DEEP-DIP( 0, 5, 0, 1)
LPEAK ( 5, 13, 32, 2)
L-DEEP-DIP( 13, 18, 4, 3)
LPEAK ( 18, 29, 32, 4)
L-DEEP-DIP( 29, 41, 7, 5)
LPEAK ( 41, 47, 115, 6)
LPEAK ( 47, 79, 31, 7)
S-DEEP-DIP( 70, 72, 0, 8)
LPEAK ( 72, 79, 37, 9)
S-HIGH-DIP( 78, 79, 12, 10)
HNS ( 82, 87, 210, 11)
S-HIGH-DIP( 88, 91, 36, 12)
LPEAK ( 91, 101, 20, 13)
S-HIGH-DIP( 101, 102, 36, 14)
LPEAK ( 102, 111, 123, 15)
S-DEEP-DIP( 114, 117, 13, 16)
LPEAK ( 117, 128, 29, 17)
S-HIGH-DIP( 128, 129, 11, 18)
LPEAK ( 129, 148, 27, 19)
L-DEEP-DIP( 148, 148, 0, 20)

```

TABLE 3.2

The Primary Acoustic Cues

Short-deep-Dip	(of the total energy)
Long-deep-Dip	(of the total energy)
Short-high-Dip	(of the total energy)
Short-medium-Dip	(of the total energy)
Mpeak	(medium duration peak)
Lpeak	(long duration peak)
Lowpeak	(low amplitude peak)
SNS	(short fricative segment)
MNS	(medium fricative segment)
LNS	(long fricative segment)
MVI	(medium sonorant segment)
LVI	(long sonorant segment)

The waveform features are sent to the next expert, the Primary Phonetic Feature Descriptor (PPFD). The PPFD generates hypotheses of phonetic features that are related to the acoustic cues with Context-Independent Rules (CIR). The descriptions made by PPFD are called the Primary Phonetic Features. The rules for generating these features are speaker-independent because they are based on perceptually significant acoustic cues. These rules are stored in the LTM of the expert. Table 3.4 shows the alphabet of these features.

Only one symbol from the above table will be generated by the PPFD for labelling and segmenting a set of acoustic cues. For example, the acoustic cue sequence,

<Long-medium-dip>|<Long-high-dip><Mpeak>

would produce the symbol, NC (non-sonorant continuant consonant), after testing and evaluating all necessary sensory procedures. In Chapter V, the intermediate output from PPFD demonstrates this process.

PPFD performs two major tasks, namely, labelling and segmenting. The segments generated by PPFD are called Pseudo Syllabic Segments (PSS). The segmentation rules are based on an attributed grammar proposed by DeMori [18]. The rules operate in a top-down fashion.

TABLE 3.4

Primary Phonetic Features

<u>Symbol</u>	<u>Feature Descriptions</u>
VF	Front vowel
VC	Central vowel
VB	Back vowel
VFC	Front or central vowel
VBC	Back or central vowel
VW	Uncertain vowel
NI	Non-sonorant interrupted consonant
NA	Non-sonorant affricate consonant
NC	Non-sonorant continuant consonant
SON	Sonorant consonant
NIV	The /v/ or a NI consonant
SONV	A sonorant or the /v/ consonant

Referring back to Fig 3.1 on the Expert System Society, there is an inter-expert communication link between the PPF and the Signal Processing Expert (SPE). The PPF repeatedly interacts with the SPE. The invocation of SPE occurs as a result of a frame instantiation in PPF. The SPE carries out various signal transformations depending on the message it receives from PPF. For example, the SPE could perform LPC analysis, FFT analysis, and so on. It is important to note that one of the novelties of this system is that it performs time-consuming signal processing analysis only when it is necessary and also only for a given frame of speech signal.

The PPF could also retrieve information stored in GSF-STM. This is necessary when the cues generated by WCD are insufficient to make a hypothesis of PSS or PPF.

A series of frame instantiations made by the PPF and SPE is shown in Appendix B-I and B-II for our test sentence.

In Appendix B-I, LTM3 is the long-term memory of PPF. The operation P-READ causes to receive a message from WCD which consists of the acoustic cue LDEEPDIP(0,5,1) and the parameters t_b , t_e , and z_c . The FR-STATE n 's are the frame instantiations.

The filled-by FRn's corresponds to the instantiation of the frame, n in SPE. Appendix B-II shows the sequence of instantiations by SPE itself.

Whenever the instantiation starts with the frame name FR-STATE 1, it means that the end of a Pseudo Syllabic Segment was reached with the current instantiation and a new syllable hypothesis could be started with the incoming cues from WCD.

A new frame instantiation is always caused by the P-READ operation . The term described-as follows a PPF which is described by the PPFD together with time duration of the feature. The PS n's are context-independent predicates.

The descriptions made by PPF are stored in the STM of the expert. Fig 3.6 shows the Phonetic Feature Descriptions generated by the system for the test sentence along with its prototypical description . The features thus obtained are sent to the syllabic expert and then to the lexical expert. A solution to the problem of accessing a large lexicon is described in detail in the next chapter.

Chapter IV

A SOLUTION TO THE PROBLEM OF ACCESSING A LARGE LEXICON

The Pseudo-Syllabic Segments generated by the Primary Phonetic Feature Descriptor which is stored in the STM of PPFD has to be used for accessing words from the lexicon. This lexicon is very large and presently it contains the 9000 most frequently used English words. Accessing such a large lexicon for real-time recognition needs some special approach so that only a small set of words will be retrieved for further recognition processes.

In this chapter we discuss syllabification of the phonetic-feature string into proper syllables, lexical organization, and finally a technique for accessing the lexicon.

4.1 Problem of Lexical Representation and Lexical Accessing.

The first problem related to lexical accessing is that the Primary Phonetic Feature Symbols generated by the system are not error free. Errors may arise due to unambiguous insertions of a wrong segment, loss of a segment, or the absence of the correct hypothesis in a segment. Word hypothesization becomes difficult as a result of this.

Another problem is that a word is not only a complex

relation between orthographic symbols and acoustic cues; it also has to contain information which is syntactic and semantic in nature [29]. This makes the lexical item a complex data structure.

Lexical accessing itself is the third problem. Two types of lexical access are known: the top-down (model-driven) and the bottom-up (data-driven). The top-down method is based on the analysis-by-synthesis model of speech perception where information drawn from phonetic features is used to search through the lexicon for possible words. A set of words is then selected and the syntactic and semantic constraints are used for hypothesizing the following adjacent word.

Considering the existing problems, a new approach has been developed which will be discussed in detail later on. In this approach, the generation and verification of the lexical hypothesis is seen as a complex problem to be solved by a complex plan. Lexical representation is obtained by a problem-reduction representation where subproblems involve the evaluation of the syllabic hypothesis and detection of acoustic cues. Each subproblem is represented by a graph of subproblems [32]. A lexical problem is solved when some of its subproblems are solved.

A degree of solution for each elementary problem is defined. Rules are provided for combining degrees of solution in order to obtain evidence for the lexical

hypothesis which is generated when the corresponding lexical problem has been solved.

4.2 Syllabification using Dynamic Programming

DP-matching techniques were used for nonlinear time-normalization in various isolated word recognition systems. The basic idea of DP-matching has been reported in several publications [20,21,22]. In syllabification, the template which is a string of phonetic features is matched against an unknown string of the same type. This unknown string may be shorter or longer in length compared to the template.

In isolated word recognition systems, DP-matching is used for time-warping and recognition by computing the minimum distance between the target word and the template. However, the concept of Dynamic Programming is used here for segmentation of phonetic feature descriptions into Pseudo-Syllabic-Segments (PSS) which contributes an alignment for learning a Lexical Access Tree.

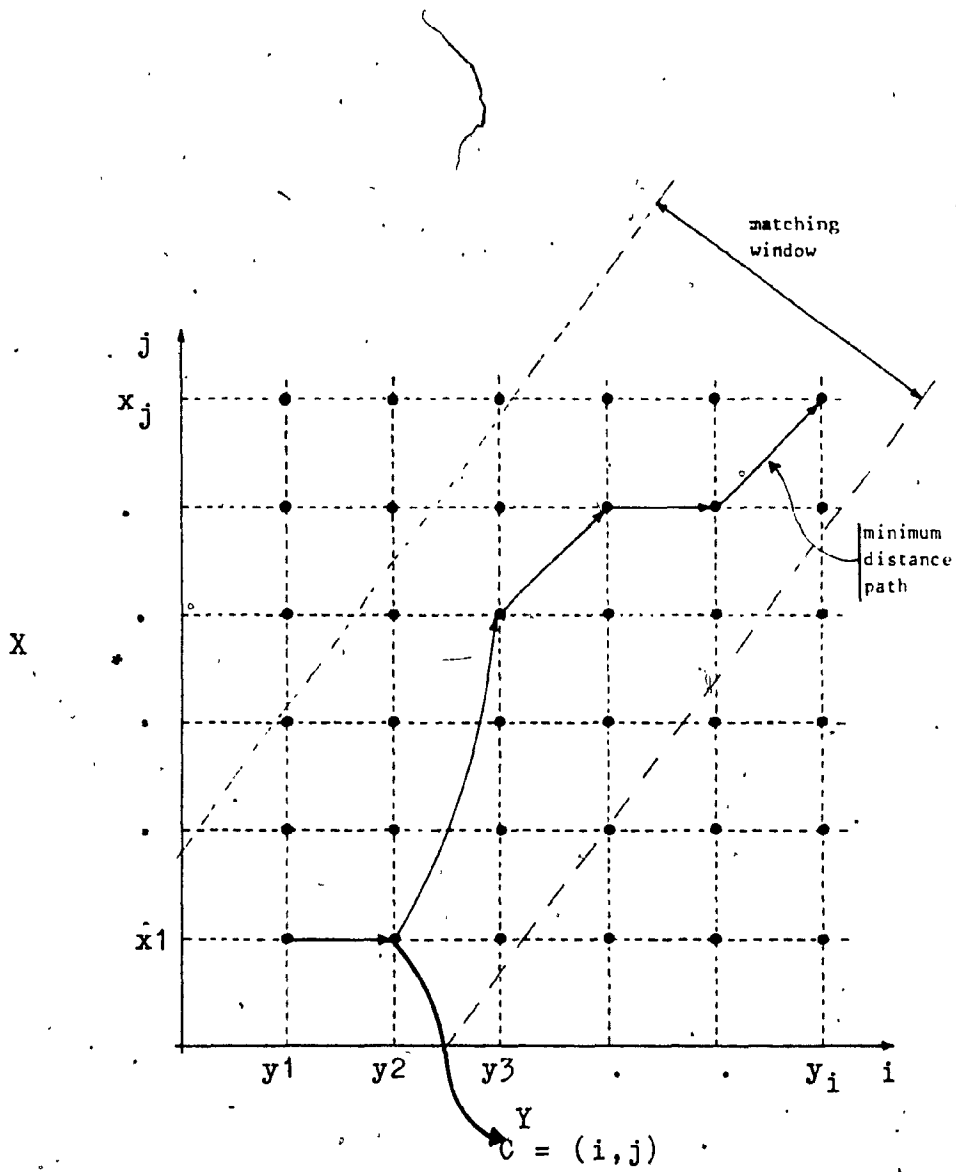
DP-Matching Principle

Let,

$$\begin{aligned} X &= x_1, x_2, \dots, x_i, \dots, x_I \\ Y &= y_1, y_2, \dots, y_j, \dots, y_J \end{aligned} \quad (1)$$

where,

Fig 4.1 DP-Matching for Time Warping



X is the speech pattern of the template (prototype) and Y is the speech pattern of the unknown word. The problem is to eliminate the timing difference between these two patterns.

If we consider an i - j plane as shown in Fig 4.1, where X and Y are developed along the i -axis and j -axis respectively, then the timing difference between the two patterns can be depicted by a sequence of points, $c=(i,j)$

$$F = c(1), c(2), \dots, c(k), \dots, c(K) \quad (2)$$

where,

$$c(k) = (i(k), j(k)).$$

The function, F, approximately realizes a mapping from the time axis of pattern Y onto pattern X. This function is called the warping function.

When there is no timing difference, that is, the points on the pattern match exactly, the warping function coincides with the diagonal line, $j=i$.

The distance between the two pattern vectors is calculated by,

$$d(c) = d(i,j) = || y_i - x_j || \quad (3)$$

Now, the total distance between the two patterns becomes,

$$F = \sum_{k=1}^k d(c(k)) \cdot w(k) \quad (4)$$

where, $w(k)$ is a weighting coefficient. The definition of the weighting coefficient $w(k)$ is given by Sakoe and Chiba [20] for both the symmetric and asymmetric form.

A practical DP-equation could be written as,

$$g(i,j) = \min \begin{cases} g(i,j-1) + d(i,j), \\ g(i-1,j-1) + d(i,j), \\ g(i-1,j) + d(i,j) \end{cases}$$

where, $g(i,j)$ is the distance computed from the origin to the point (i,j) .

This typical DP-matching technique is based on the following conditions:

- a) Speech patterns are time-sampled with a common and constant sampling period.
- b) There is no a priori knowledge about any part of the speech sample, such as linguistic information.

The initial condition, $g(1,1) = d(1,1)$ holds.

- c) The window adjustment is defined as,
 $j-r \leq i \leq j+r$, where r is the window size
- d) The domain in which the DP-equation is calculated is,

$1 \leq i \leq I$ and $1 \leq j \leq J$ where, I and J are the length of the speech patterns X, Y .

The DP-equation described above was modified to meet our objective in order to learn an acceptable degradation for the speech pattern. The condition (b) does not hold any more since the input speech patterns are different in nature as opposed to the data used for recognition purposes.

First, phonetic feature vectors are the input data for the DP-matching system. X is the template feature vector, and Y is a degradation of X generated by the system. Since these patterns contain linguistically important information, the weighting coefficient $w(k)$ is selected with an a priori knowledge.

To satisfy these characteristics, the DP-equation (5) is modified as,

$$g(i,j) = \min \begin{cases} g(i,j-1) + D_i, \\ g(i-1,j-1) + S_{ij}, \\ g(i+1,j) + I_j \end{cases} \quad (6)$$

where,

D_i is the deletion error (weight).

S_{ij} is the substitution error, and

I_j is the insertion error.

Instead of one weighting coefficient, there are 3 $w(k)$'s associated.

Fig 4.2 Example of Deletion, Insertion, and Substitution

Proto
type

PP
 NI
 SON
 VF
 SON
 NI
 VB
 SON
 VF
 NC
 VF
 NI
 NUL

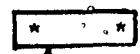
NUL NIP VBC SONP VBC SONP VBC NIP VFPC NC PP

Deletion

Substitution

Insertion

Unknown



Deletion Error (D_i)

The deletion error is the deletion of a feature on the prototype. The distance path travels vertically up if a deletion occurs. The weight varies depending on which symbol has to be deleted. For example, the deletion error for a vowel is ten where as, the error for deleting a vowel in a diphthong is only three. Fig 4.2 shows an example of a Deletion, Insertion, and Substitution errors.

Insertion Error (I_j)

Insertion error is the insertion of a symbol in the unknown speech pattern, Y. The insertion error of a SON is five and ten for a consonant. The path travels horizontally upon an insertion.

Substitution Error (S_{ij})

Substitution is the lowest if the symbols match exactly. For example, if a vowel matches with a vowel, the error is 0. If this occurs then we have no degraded symbol. The path goes diagonally after substitution. The substitution weights (error) vary for various categories of features and are kept on a look-up table. The substitution error for a vowel with a consonant is 10, which is the maximum.

Fig 4.3 DP-Matching with Added Weight Functions.

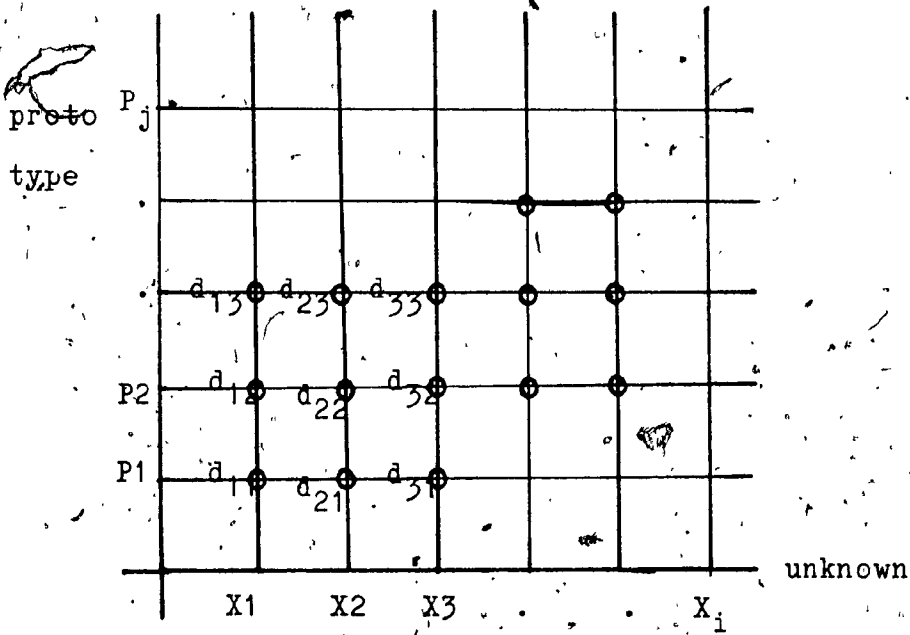


Fig 4.3 shows an illustration of the matching technique. As an example, the distance d_{22} can be calculated as,

$$d_{22} = \min \begin{cases} d_{21} + D(P_2), \\ d_{11} + S(X_2, P_2), \\ d_{12} + I(X_2) \end{cases}$$

The minimum distance points are located and the path is drawn from the origin to the end.

The syllable bounds of Y (the unknown string) can be found by drawing a perpendicular line to the i-axis (Y-vector) from the point of intersection between the minimum distance path and the syllable bound line on the j-axis (X-vector or the prototype). In Fig 4.4, S_p is the syllable-bound-line on the prototype and S_u is the syllable-bound-line found on the unknown after performing the DP-matching algorithm.

The window size, r , is variable and the value between 4 and 6 gave a very good result. Fig 4.4 shows a system generated DP-matching path for the unknown word "development" and its prototype.

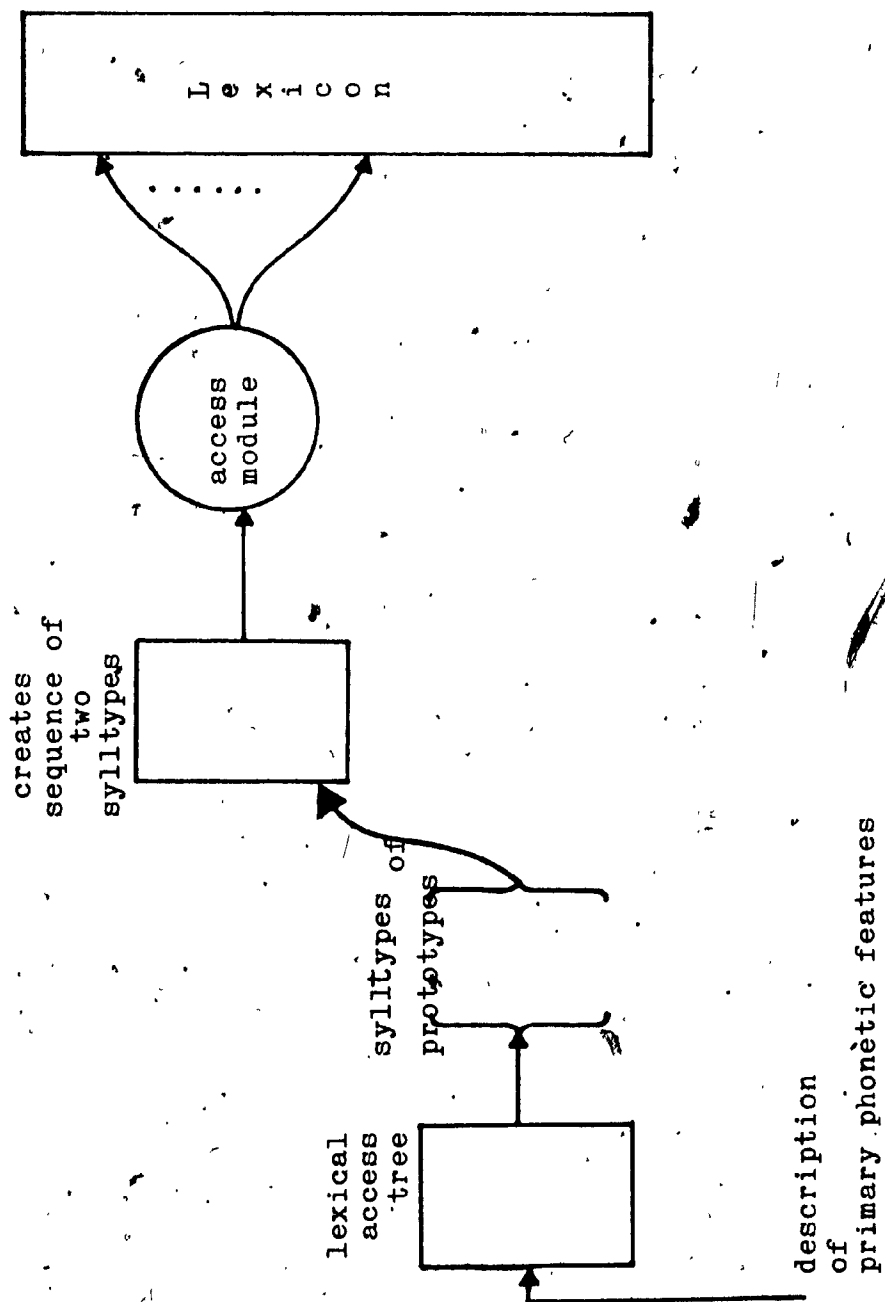
The syllables thus obtained are Pseudo-Syllabic-Segments of Y, the unknown. These syllable segments are used to learn a syllabic-tree (word-access-tree).

4.3 Lexical Access

The syllabic segments obtained from DP-matching are the main tools for accessing the lexicon. A word hypothesization is not possible at this point. The syllable segments simply contain a degraded version of primary phonetic features which will correspond to a large set of words in the lexicon. In other words, there may be many words which have the same phonetic structure.

The objective is to access a minimum number of such words from the lexicon and make a word hypothesis. The system under consideration has a structure as shown in Fig 4.5. A detailed study of the function and implementation of this system components is now considered.

Fig 4.5 A Lexical Access Model



The Lexical Access Tree

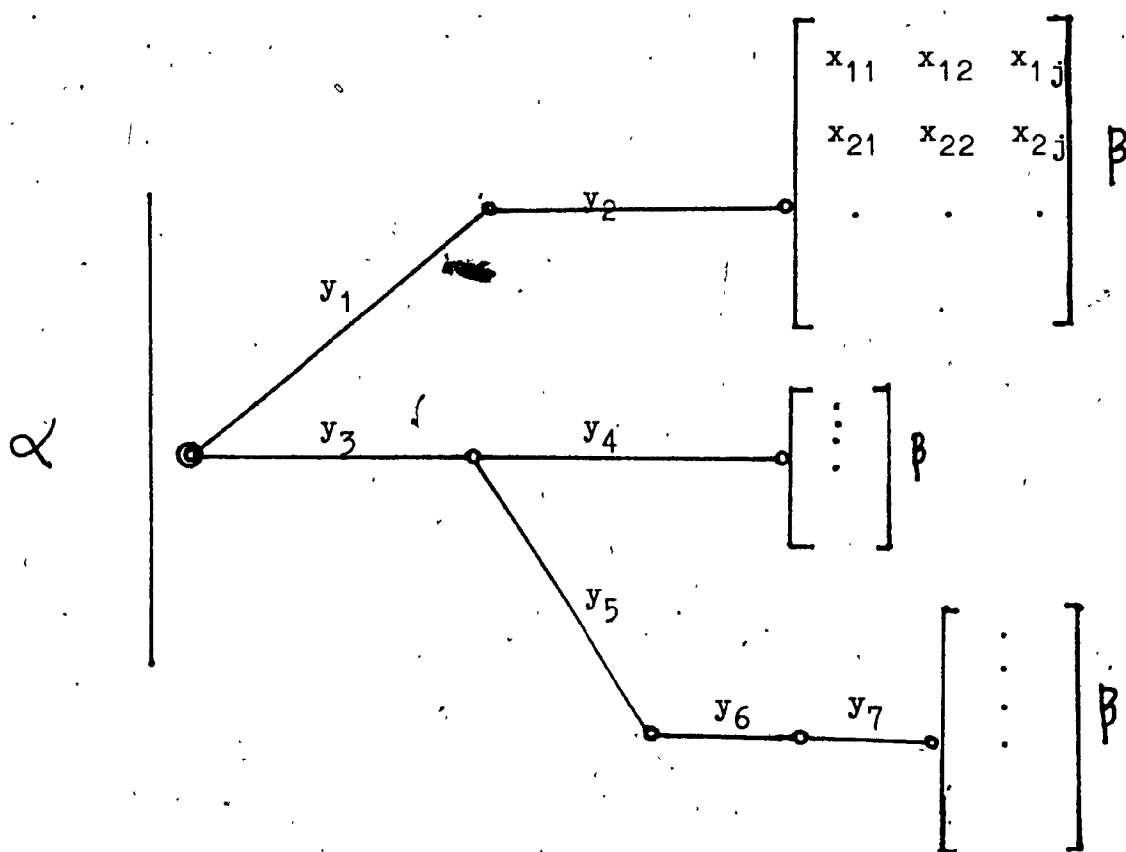
During the learning stage of the system, several words will be spoken by several people and the system will generate the degraded version of syllabic segments. We consider only the first two syllables of each word. The syllable segments generated after DP-matching are used to grow the word access tree. This tree has a structure shown in Fig 4.6.

In Fig 4.6, Y_i Y which is the alphabet of phonetic features detected with context-independent rules and x_i X , which is the prototype.

Each node of this tree contains one phonetic feature. The leaf node has more information stored in it. It contains the prototype of that particular syllable, the stress information, the frequency count of the number of times that particular node has been accessed, and also the facility of accommodating more information, if necessary, using the same path.

During the learning process, the phonetic descriptions are sent to a Lexical Access Tree (LAT) prior to performing the DP-matching algorithm. The LAT could be updated in two different ways: by creating a new node with a current symbol or by adding a new entry into the leaf of any given node. Three actions are possible whenever the tree is accessed:

Fig 4.6 Model of Lexical Access Tree



α is a degradation of β .

- i) the tree is updated by creating a new node.
- ii) the leaf of a node is updated
- iii) no updating of the tree. A set of words is accessed from the lexicon

The LAT algorithm starts with the phonetic symbol described at time, $t=0$. With this symbol, the tree is searched starting from the root. If any node matches with the symbol then the following branch is searched for the next symbol at $t = t+1$. The search will terminate if any of the following three conditions are satisfied:

- i) if no more symbols are left on the PFFD's symbol sequence
- ii) if the branch node = nil
- iii) if the symbol does not match.

If termination occurs (this must happen), all the syllable entries at the leaf of the node at time $t-1$ are used to access the lexicon.

The tree will saturate at some point during the learning process since the maximum number of feature combinations to make a permissible syllable is less than 100.

In Fig 4.6, x_i 's are the features in the prototype and y_j 's are the features of the unknown. In other words, we

say, y_j 's are degradations of x_i 's. The stress information is represented as s , and f is the frequency count.

4.4 Organization of Lexical Knowledge

The words of a lexicon and their relations with syllables, acoustic, and prosodic cues, are represented using a semantic network model [19]. We call this a Lexical Network. The main components of a Lexical Network are nodes and links. Each node is associated with a name, a body of knowledge, and a set of procedures. Links establish relations between nodes and have associated descriptions of the relations. The types of messages which could be sent between the nodes depend on these relations.

The Lexical Network is described by a graph grammar in which nonterminal symbols are represented by strings of lower-case letters and terminal symbols with upper-case letters. These rules are used for word hypothesization and lexical accessing. Some of the rules are given below with the starting symbol as "lexicon".

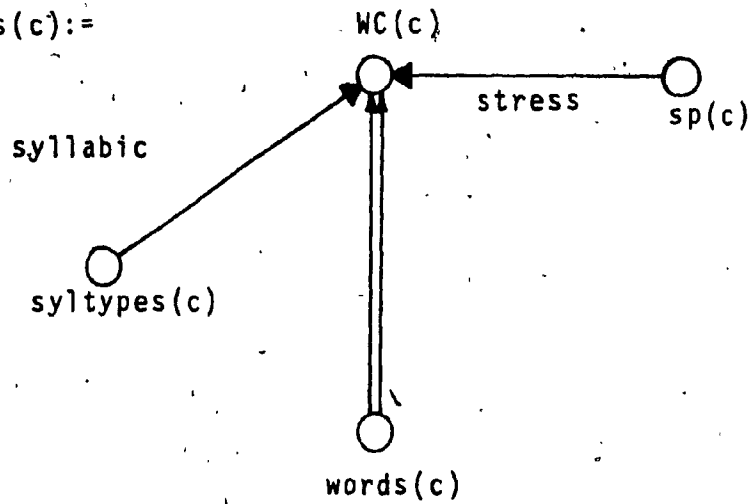
Rule RL1

$$\text{lexicon} := \text{wcs}(1) | \text{wcs}(2) | \dots | \text{wcs}(c) | \dots | \text{wcs}(g)$$

The symbol '|' represents a disjunction of items which can be generated by the nonterminal symbol 'lexicon'.

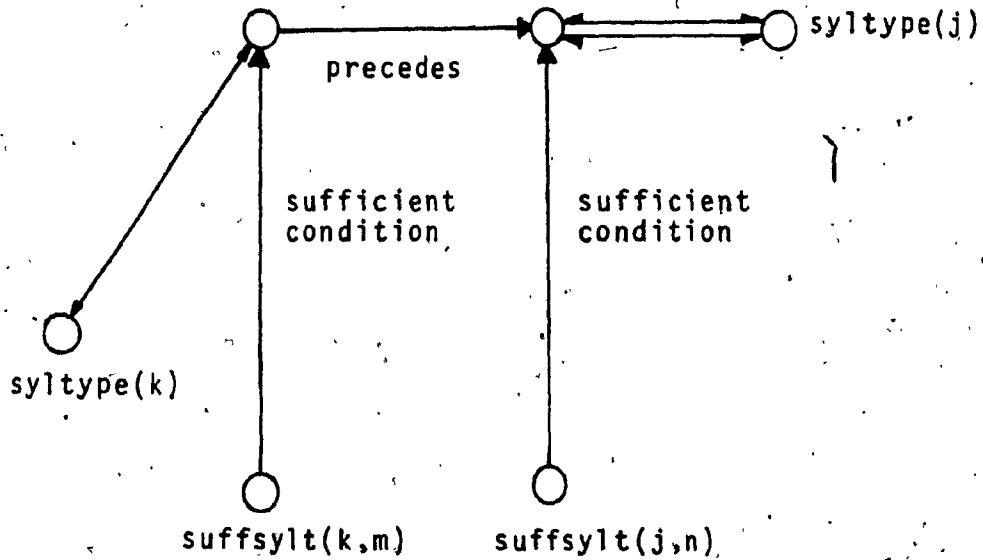
Rule RL2

wcs(c) :=

Rule RL3

syltypes(c) := SYLT(c,1)

SYLT(c,2)



According to the Rule RL1, the Lexical Network is a collection of structures corresponding to word classes. Each word in the lexicon may belong to one or more word classes $WC(c)$ depending on the variations it may get. $WC(c)$ is the label of the node after further re-writing the rule RL1.

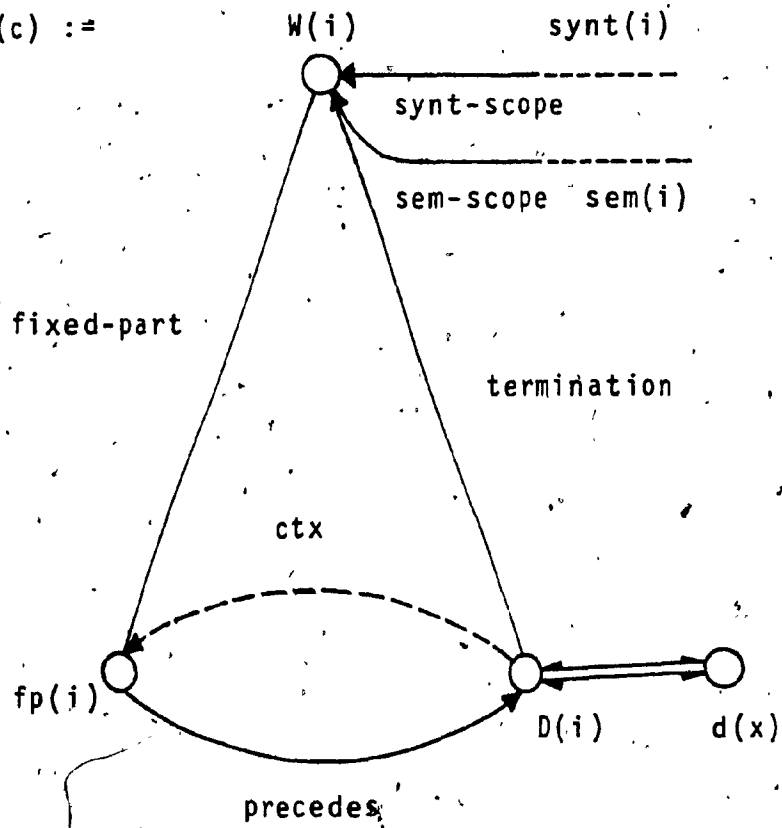
A word class $WC(c)$ contains some stress information and a sequence of syllable types. A virtual copy link connects the node $WC(c)$ with the words belonging to that class as shown in the graph of rule RL2. A word class is activated when some sufficient conditions are met. This property of the word class will be inherited by every word node and become a virtual copy of the word class. Therefore, the nonterminal symbol $words(c)$ are used to represent the virtual copies of $WC(c)$.

The nonterminal symbol, $sylltypes(c)$ in Rule RL2 is used for generating the phonetic features of the syllable belonging to the word class $WC(c)$. This symbol is further re-written in Rule RL3.

According to Rule RL3, the $sylltypes$ of word class $WC(c)$ are represented by two nodes, $SYLT(c,1)$ and $SYLT(c,2)$. The link from $SYLT(c,1)$ to $sylltype(k)$ means that the first syllable type $WC(c)$ is equivalent to the structure of $sylltype(k)$. $SYLT(c,2)$ has the same structure as that of $sylltype(j)$.

Rule RL5

$W(c) :=$



SYLT(c,1) precedes SYLT(c,2) in time. SYLT(c,1) and SYLT(c,2) are linked with a set of sufficient conditions suffsylt(k,m) and suffsylt(j,n), respectively. The nodes SYLT(c,1) and SYLT(c,2) become active only if these sufficient conditions are met.

The word class WC(c) becomes active when all the sufficient conditions are met and the relations between them are verified and also when the stress information sp(c) is satisfied and syltypes(c) are active. When WC(c) becomes active the nonterminal symbol words(c) are expanded using Rule RL4.

Rule RL4

$$\text{words}(c) := W(c,1) | W(c,2) | \dots | W(c,i) | \dots | W(c,I)$$

Ic is the number of items. Each word W(c,i) of the word class WC(c) inherits the properties of that particular word class and has a node W(i) for describing its orthographic representation. W(i) also possesses information such as syntactic and semantic descriptions. Links are also established between the node W(i) and the fixed part of W(i) as shown in Rule RL5. In Rule RL5, the node fp(i) precedes the node D(i). D(i) is the termination node of the word. D(i) will contain a set of prototype terminations. The dashed link between D(i) and fp(i) called 'ctx' means that phonemes in D(i) may act as context constraints in the rules relating phonetic features of fp(i) with acoustic cues.

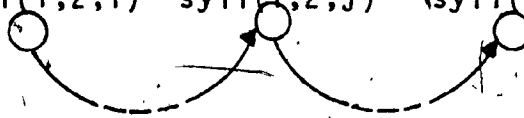
Rule RL7

$fp(i) ::= \text{syll}(i,1,1) \text{ syll}(i,1,j) \text{ syll}(i,1,J(ij))$



precedes precedes

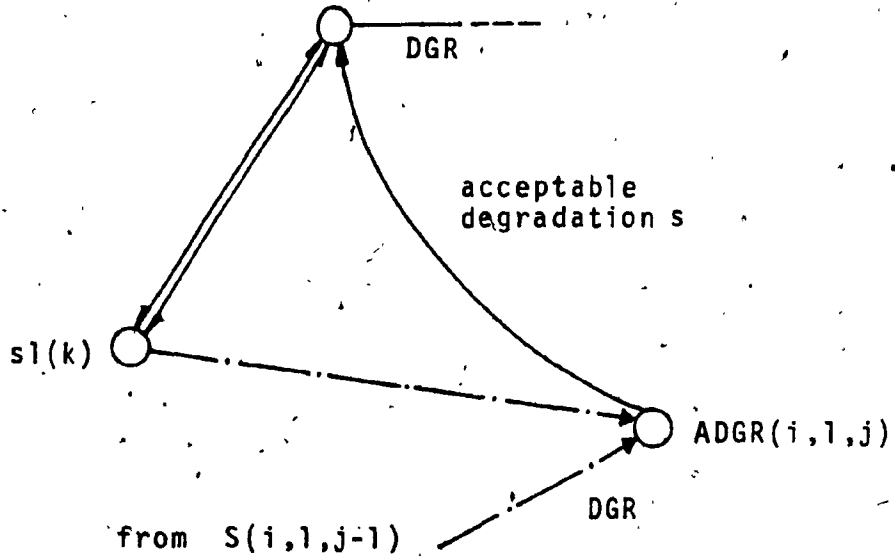
$:= \text{syll}(i,2,1) \text{ syll}(i,2,j) \text{ syll}(i,2,J(i2))$



precedes precedes

Rule RL8

$\text{syll}(i,1,j) ::= S(i,1,j) \dots \text{to } S(i,1,j+1)$



The nonterminal symbol $d(x)$ could be written as shown in Rule RL6.

Rule RL6

$$d(x) := \text{DSINGULAR}(x) \mid \text{DPLURAL}(x)$$

The termination parts of words are different for plurals and singulars, which affects the last syllable. For this reason, the fixed part and termination part are separated.

Rule RL7 establishes that the nonterminal symbol $fp(i)$ is a disjunction of references of syllables. The last syllable of each reference is usually incomplete and has to be completed by the termination part. The structure of each syllabic node is defined by Rule RL8.

Each syllable $S(i,1,j)$ is an instantiation equivalent to a syllable represented by the node $sl(k)$. The node $S(i,1,j)$ is activated if a set of acceptable degradations of $sl(k)$ has been hypothesized from the data. The set of acceptable degradations is contained in the node $\text{ADGR}(i,1,j)$ which initially receives them from $sl(k)$ through the link DGR which could be modified if necessary.

Fig 4.7 Algorithm to Access Large Lexicon

```
begin  
for every c do  
  begin  
    apply-rule-R12(wcs(c))  
    apply-rule-R13(sylltypes(c))  
    matchphon(SUF(c),data(Ta,Tb), var match:boolean)  
    if match then  
      expand-node(WC(c))  
    end  
  end  
end
```

The sufficient conditions for the word class, $SUF(c)$, is the primary phonetic feature vector of the syllable detected. Sequences of primary phonetic features define syllable-types characterizing classes of syllables. The total number of such classes is less than 100. An algorithm for accessing a large lexicon using the features generated by the system in the time interval $T(a,b)$ is shown in Fig 4.7.

The algorithm shows that many subnetworks corresponding to word classes for which there is sufficient evidence in the data can be expanded in parallel. Global evidences are computed only for those word hypotheses belonging to pre-selected word classes, that are consistent with top-down predictions and for which enough phonetic features have been detected in the data.

Chapter V

SYSTEM PERFORMANCE AND EVALUATION

The Expert system explained in the previous chapters is successfully implemented and currently running on a VAX 11/780 machine of the Graphic Interactive Laboratory at Concordia University. The implementation was done in Pascal language, however, work is already under way to transfer the experts into LISP language which is more efficient for such applications.

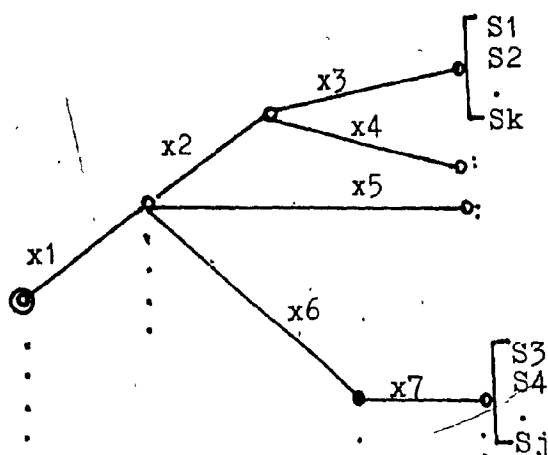
The speech signal was sampled at 20 KHz and quantized over 12 bits in a normal computer room environment.

The orthographic representation of the 9000 most frequently used English words with phonemic code was created and used as the lexicon of the system. With every word in the lexicon, information such as total number of syllables and stress information were detected and attached automatically.

5.1 Experiments on Lexical Accessing

The prime objective of this part of the work was to investigate the size of the space of processes corresponding to nodes of the type $w(c,i)$. A second objective was to test the speaker-independent capability of the system.

Fig 5.1 Algorithm for Sufficient Conditions



In order to verify the first objective, the Lexical-Access-Tree was learned first and access rates were collected. Sufficient Conditions (SC) were defined in order to execute the LAT algorithm.

Choice of Sufficient Conditions

Different types of SC's could be considered. The SC which was chosen and implemented has a structure as shown in Fig 5.1. Every x_i in the node is a phonetic feature generated by the system. The entries in the first leaf node in the tree, while searching, are considered as the possible first syllables (sylltype(k)) of the word and the entries in the next leaf node as the second syllable (sylltype(j)). In Fig 5.1, the $S_{k,j}$ are the sylltypes.

The words with sylltype(k) succeeded by sylltype(j) are accessed from the lexicon. The total number of words thus selected would be of the syllable combinations, as shown below.

$$[(S1 S3), (S1 S4), \dots, (S1 Sj), (S2 S3), (S2 S4), \dots, \\ (S2 Sj), \dots, (Sk Sj)]$$

Another choice of SC would be to perform DP-matching on every sylltype in the leaf node against the degraded syllable sequence obtained from the tree. Lexical access will be made only if the distance between the syllables is below a certain threshold. The following algorithm establishes the lexical accessing.

```

begin
  for every  $S_{d1}$  do
    begin
      perform-DP-matching( $S_{d1}, S_1, \text{distance1}$ )
      if ( $\text{distance1} < \text{threshold}$ ) then
        perform-DP-matching( $S_{d2}, S_2, \text{distance2}$ )
        if ( $\text{distance2} < \text{threshold}$ ) then
          expand-node( $WC(c)$ )
        end
      end
    end
  end

```

where,

$S_{d1}, d2$ are degradations of S_1 and S_2 .

At the expense of more computing time, this choice of SC will extract less words from the lexicon in comparison with the previous one. Ambiguous word selection could also be avoided.

The choice of SC in the first case is obviously the worst case yet the system accessed 2.3% of the lexicon which is well within the acceptable range. Another important aspect of LAT is its capability for learning. The percentage of learning of the LAT must go down as more words are learned and finally it must saturate. An experiment was conducted to verify this.

Fig. 5.2a Output of the Expert PPF

```
( 0, 6, 0, 1 LDEEPDIP)
( 6, 16, 30, 2 LPEAK)
( 16, 17, 22, 3 SHIGHDIP)
( 17, 26, 30, 4 LPEAK)
( 26, 28, 32, 5 SHIGHDIP)
( 28, 35, 40, 6 MPEAK)
( 35, 48, 7, 7 LDEEPDIP)
( 48, 58, 19, 8 LPEAK)
( 58, 65, 4, 9 LDEEPDIP)
( 65, 68, 2, 10 SPEAK)
( 68, 68, 0, 11 LDEEPDIP)
```

```
INPUT_SYMBOL  CURRENT_STATE  PROCEDURE_#
-----
SHIGHDIP 16          1          4
-- NO SYLLABLE POSSIBLE --
-- NEXT STATE IS 127
```

```
INPUT_SYMBOL  CURRENT_STATE  PROCEDURE_#
-----
LPEAK 17          27          2
```

```
INPUT_SYMBOL  CURRENT_STATE  PROCEDURE_#
-----
LDEEPDIP 0          1          24
-- NO SYLLABLE POSSIBLE --
-- NEXT STATE IS 2
```

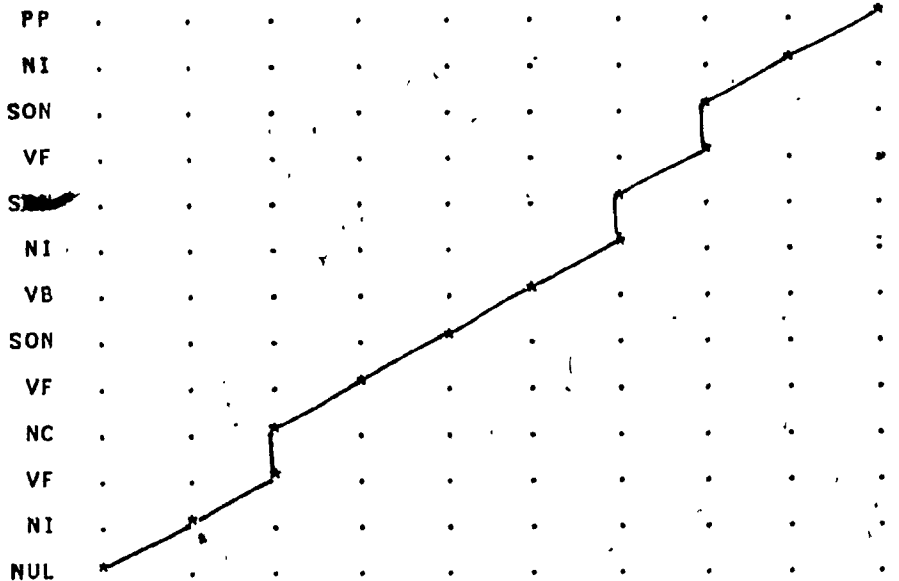
```
XXXXX DIPHTONG IS CALLED
IIIII LPC IS CALLED
-- PERFORM " OUTPUT-SYLLABLE (1) " --
-- NEXT STATE IS 1
```

```
INPUT_SYMBOL  CURRENT_STATE  PROCEDURE_#
-----
LPEAK 6          2          8
```

```
INPUT_SYMBOL  CURRENT_STATE  PROCEDURE_#
-----
SHIGHDIP 26          1          4
```

```
????? POSSIBLE BURST IS CHECKED.
XXXXX DIPHTONG IS CALLED
IIIII LPC IS CALLED
-- PERFORM " OUTPUT-SYLLABLE (1) " --
-- NEXT STATE IS 1
```

```
-- NO SYLLABLE POSSIBLE --
-- NEXT STATE IS 27
```

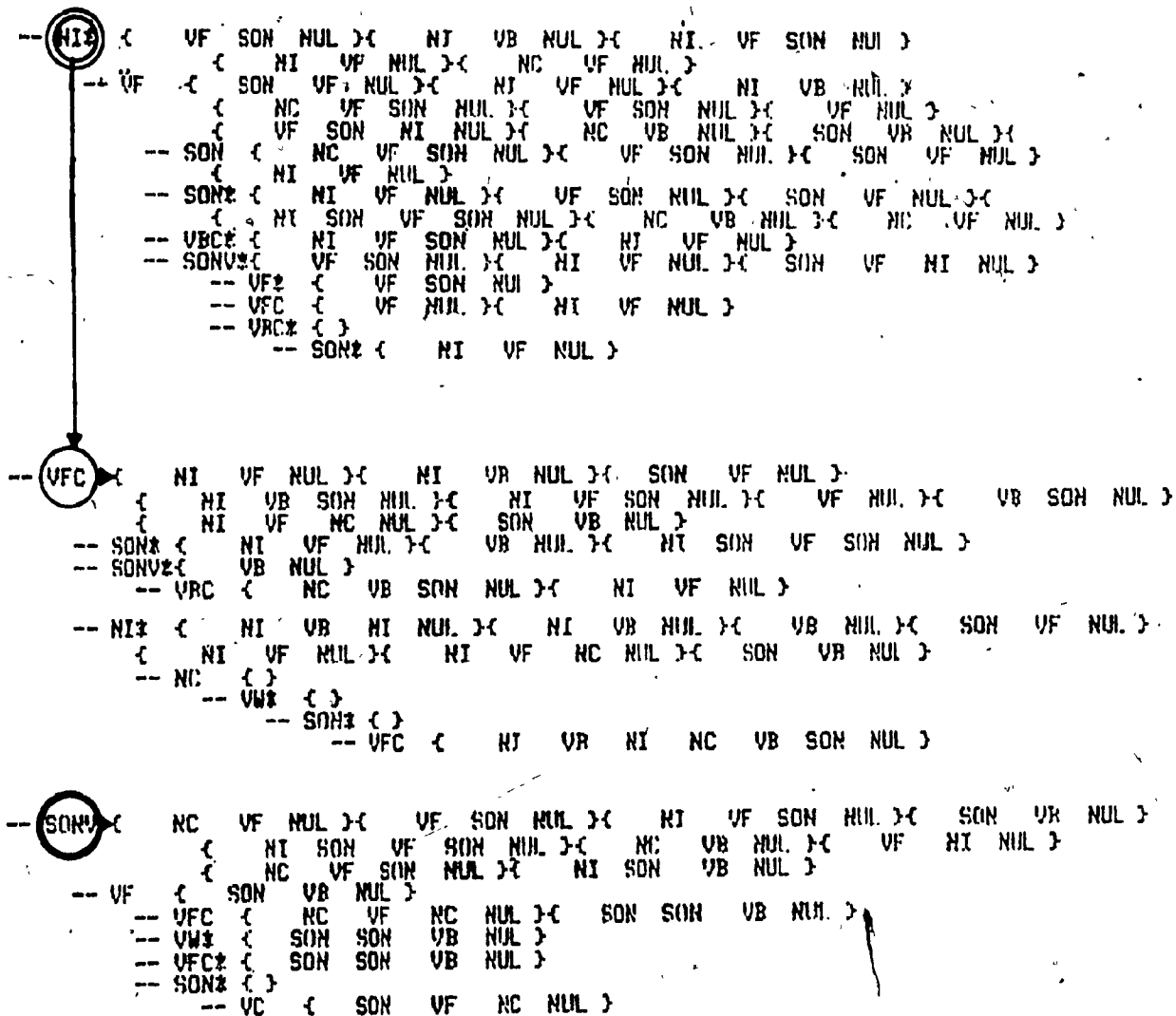


THE PATH IS: (0, 0, 1, 1, 2, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 6, 8, 7, 9, 7, 10, 8, 11, 9, 12,)

TOTAL DISTANCE = 24

**** Word accessed ! Tree was not learned ****

Fig 5.2b Section of the Lexical-Access-Tree



Accessing the word "development"

Ten sets of training words were selected at random with each set having 10 words. A single speaker was asked to utter every word in each training set and statistics were collected for all the tree updates, leaf updates, and words accessed without the need for any updates.

Fig 5.2a shows different intermediate outputs generated by various Experts during the learning stage of the word "development". In the DP-matching graph, the prototype and the system generated PPF's are shown. Fig 5.2b shows a part of the Lexical-Access-Tree with the path (drawn in thick ink) where it accesses the word correctly. The entries inside the curly brackets are the sylltypes.

Fig 5.3 shows the graph of the learning percentage versus the number of trials.

It is interesting to note, in Fig 5.4, that the number of tree updates went down rapidly as the tree started learning. Fig 5.3 also shows that as the tree learns, the number of trials needed is less for a new set of words.

From the various graphs presented, it can be concluded that the system performs within what was expected. It is also important to note that there is a lot of room for improvement in various levels of the system.

Fig 5.3 Graph of Tree Learning %

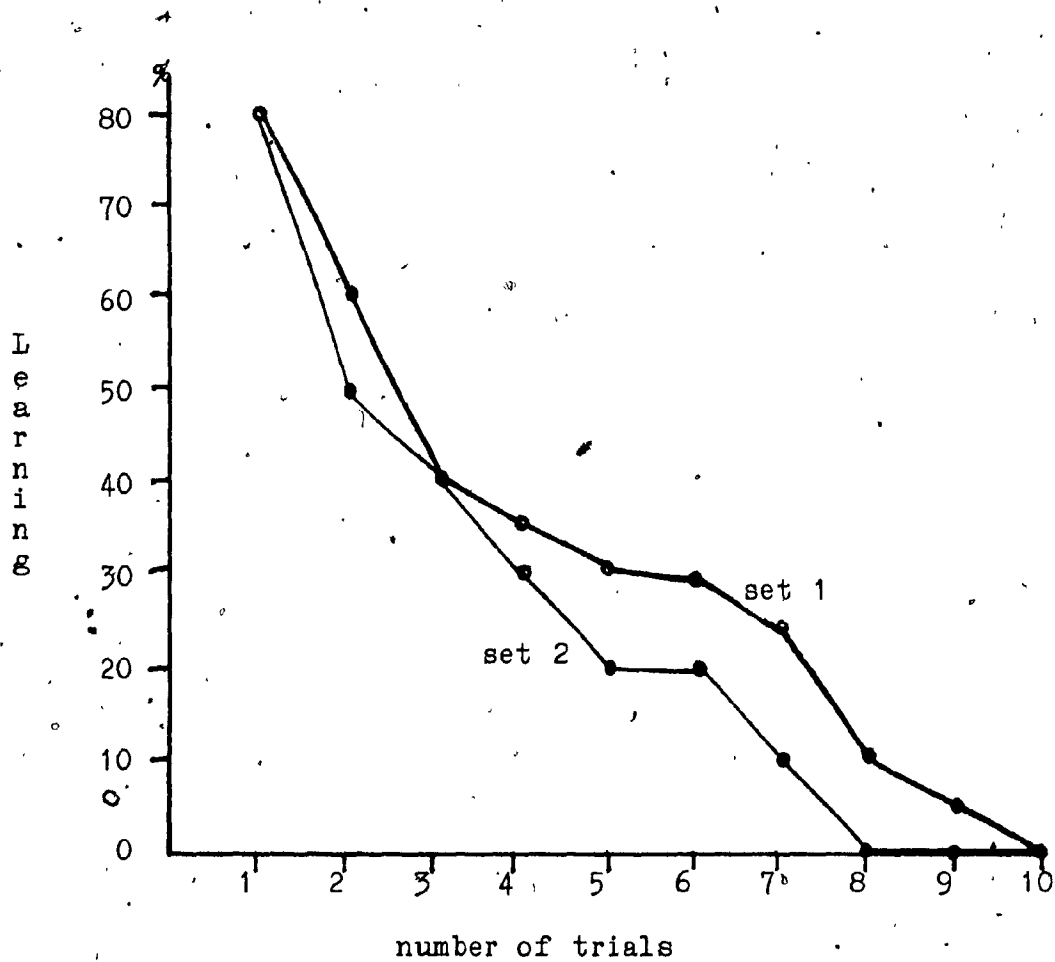
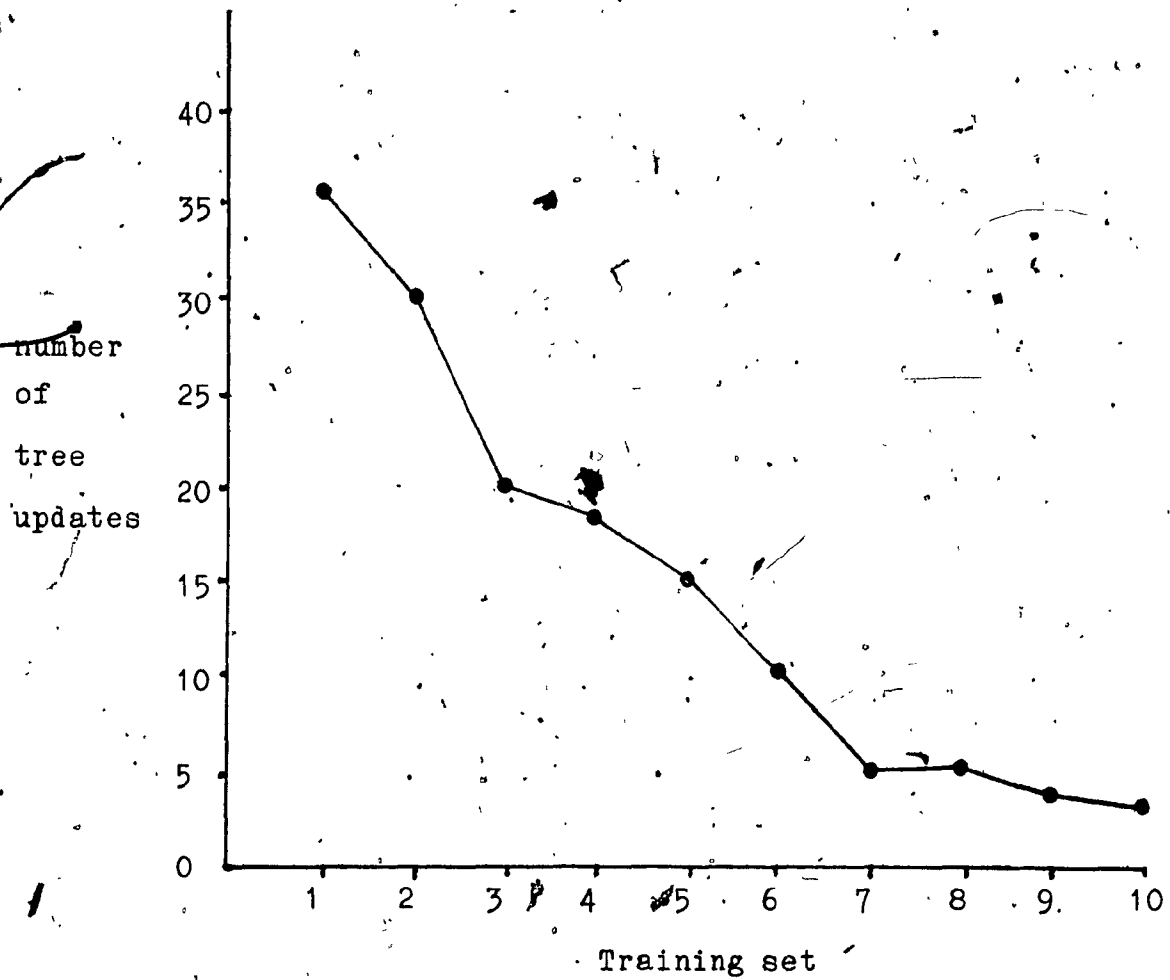


Fig 5.4 Graph of Tree Updates



5.2 Conclusion

Various design objectives were achieved satisfactorily. It was shown that word hypothesization need not be done on a large lexicon but only on a subset. Top-down constraints can now be applied for further reducing the size of the subset into a still smaller percentage of the lexicon.

Some of the different areas where more work is needed to improve the performance of the system are:

i) improving the PFFD expert to generate better phonetic feature descriptions. For words with stressed syllables the descriptions are very good. However, for words starting with certain sounds, the system either misses or generates ambiguous symbols. These errors could easily be corrected by developing more rules and executing sensory procedures.

ii) more heuristics are needed while learning the tree. For example, spurious phonetic sequences which are at times generated by the PFFD should not be used to learn the tree. These must be detected automatically and rejected accordingly.

iii) More Sufficient Conditions are to be considered for accessing the lexicon. Since these conditions affect the percentage of words being

accessed from the lexicon, much more work has to be done.

iv) Further elimination of word hypothesization can be done with a controlled extraction of detailed features in well-specified intervals and a well-specified context. This might need further execution of very sophisticated signal processing and feature extraction algorithms which need not be executed all over the signal but only for the specific interval.

v) Real-time performance of the system cannot be tested since a timesharing system is currently being used. The response time, however, is still good; approximately 3 minutes turnaround time to access the words.

vi) Special purpose computer architecture is needed to achieve parallelism. The system was conceived in such a way as to meet these requirements whenever multiprocessor systems are available. This is also one of the qualities of the expert systems.

vii) Speaker-independency was also successfully demonstrated. The system correctly hypothesised words spoken by non-native English speakers who had very heavy accents.

REFERENCES

1. Speechlab Laboratory Manual, Heuristics Inc., Los Altos, CA, Vol 1, 1977
2. S. E. Levinson and M. Y. Liberman, "Speech Recognition by Computer", Scientific American, vol. 4, April 1981, 64-74
3. R. M. Schwartz, "Acoustic Phonetic Recognition", 6th International Conference on Pattern Recognition, Munich, 1982, 952-965
4. V. W. Zue and R. M. Schwartz, "Acoustic Processing and Phonetic Analysis", Trends in Speech Recognition, R. W. Lea (Ed.), 1980, 101-125
5. V. W. Zue, "Acoustic Characteristics of Stop Consonants: A Controlled Study", Sc.D. thesis, Mass. Inst. of Tech., Cambridge, MA, 1976
6. P. Mermelstein, "Computer Recognition of Continuous Speech", Computer Analysis and Perception of Visual and Auditory Signals, C. Y. Suen and R. De Mori (Ed.),
7. R. De Mori, P. Laface, E. Piccolo, "Acoustic Detection and Description of Syllabic Features in

- Continuous Speech", IEEE Transactions on Acoustic Speech and Signal Processing, Oct. 1976, 365-379
8. Fujimura and Osamu, "Analysis of Nasal Consonants", Journal of the Acoustic Society of America, Vol. 34, No. 12, 1962, 1865-1875
 9. P.B. Denes and E. N. Pinson, "The Speech Chain", Doubleday, Garden City, New York, 1973
 10. G. R. Doddington and T. B. Schalk, "Speech Recognition: turning theory to practice", IEEE Spectrum, September 1981, 26-32
 11. B. E. Pay and C. R. Evans, "An Approach to Automatic Recognition of Speech", Int. J. Man-Machine Studies, vol. 14, 1981, 13-27
 12. R. De Mori, Y. Mong, M. Palakal, C. Y. Suen, "Network System for Generating Syllabic Hypothesis in Continuous Speech Recognition in a Speaker-independent Environment", International Conference on Systems, Man, and Cybernetics, Bombay, 1983
 13. W. B. Gevarter, "Expert Systems. Limited but Powerful", IEEE Spectrum, vol. 19, Aug. 1983, 39-45
 14. Miller, Pople, Myers, "Internist-1, an Experimental Computer-based Diagnostic Consultant

- for General Internal Medicine", The New England Journal of Medicine, Vol 307, Aug. 1982, 468-76
15. M. Goldberg, G. Karam, M. Alvo, "A Production Rule-Based Expert System for Interpreting Multi-Temporal Landsat Imagery", Computer Vision and Pattern Recognition Conference, Washington, 1983, 77-83
 16. R. De Mori, A. Giordana, P. Laface, L. Saitta, "Parallel Algorithms for Syllable Recognition in Continuous Speech", to appear in IEEE Transactions on Pattern Analysis and Machine Intelligence.
 17. M. Minsky, "A Framework for Representing Knowledge", Psychology of Computer Vision, P. Winston (Eds.), McGraw-Hill, 1975
 18. R. De Mori, A. Giordana, P. Laface, "Speech Segmentation and interpretation using a semantic syntax-directed translation", Pattern Recognition Letters, Dec. 1982, 121-124
 19. S. E. Fahlman, "NETL: A System for Representing and Using Real-world Knowledge", The MIT Press, Cambridge, 1977
 20. H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", Automatic Speech and Speaker Recognition, N. R. Dixon and T. B. Martin

(Eds.), IEEE Press, 1979

21. V. M. Velichko, N. G. Zagoruyko, "Automatic Recognition of 200 words", Int. J. Man-Machine Studies, vol. 2, June 1970, 223
22. _____, "A Dynamic programming approach to continuous speech recognition", Proceedings of the 7th ICA, Aug. 1971
23. D. C. Levinson, "The Well-Tempered Speech Recognizer", Ph.D thesis, University of Calgary, 1982
24. J. B. Kuipers, "A Frame for Frames: Representing Knowledge for Recognition", in Representation and Understanding, Bobrow and Collins(ed.), Academic Press.
25. L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterance", The Bell System Technical Journal, Vol. 54, Feb. 1975, 297-315
26. J. T. Tou, R. C. Gonzalez, "Pattern Recognition Principles", Addison-Wesley Publishing Company, 1974.
27. G. M. White and R. B. Neely, "Speech Recognition Experiments with Linear Prediction, Band-Pass Filtering and Dynamic Programming", IEEE

Transactions on Acoustic Speech and Signal Processing, vol. ASSP-24, April 1976, 183-188

28. R. De Mori, "Automatic Speech Recognition", Applications of Pattern Recognition, K. S. Fu (Eds.), CRC Press, 1982
29. R. De Mori, "Computer Models of Speech Using Fuzzy Algorithms", Plenum Press, New York, 1983
30. J. L. Flanagan, "Speech Analysis, Synthesis, and Perception", Springer-Verlag, Berlin, 1972 (II ed.)
31. D. R. Reddy, "Speech Recognition by Machine: A Review", Automatic Speech and Speaker Recognition, N. R. Dixon and T. B. Martin (Eds.), 1979
32. R. De Mori, Y. Mong, M. Palakal, "A Network for the Recognition of a Large Spoken Vocabulary", Computer Vision and Pattern Recognition Conference, Washington, 1983, 83-89

RECENT ACHIEVEMENTS IN ISOLATED WORD RECOGNITION

<u>Institution</u>	<u>Country</u>	<u>Research</u>
University of Kyoto (OKOCHI, SAKAI, 1982)	Japan	Trapezoidal DP matching
Fujitsu (NARA et AL., 1982)	Japan	Model of typical distortions for DP matching
KTH-Stockholm (ELENIUS et AL., 1982)	Sweden	Use of temporal constraints
Bell Lab (R. BROWN, 1982)	U.S.A.	Beam search techniques
Helsinki University of Techn. (KOHONEN et AL., 1980)	Finland	Associative memories
Hewlett-Packard (GREER et AL., 1982)	U.S.A.	Beam search
Royal Signal and Radar Establishment (MOORE et AL., 1982)	England	Use of fuzzy algebra
Bell-Northern (MERMELSTEIN, 1982)	Canada	Comparison of several acoustic parameters for DP matching
Purdue University (KASYAP, 1979)	U.S.A.	Stochastic model
IBM Yorktown (SILVERMAN et DIXON, 1980)	U.S.A.	Use of acoustic constraints in DP matching
Bell Labs (BROWN et RABINER, 1982)	U.S.A.	Use of graph theory methods for matching
University of Kiev (VINTSJUK, 1980)	URSS	Use of symbols describing phones in DP-matching

SPEECH SYNTHESIS

<u>Institute</u>	<u>Country</u>	<u>Research</u>
CIT Challet et AL.	France	Speech segments
Cornell University Hertz	U.S.A.	Interactive System for rule development
KTH Carlsson et AL.	Sweden	Multi-language speech synthesizer
MIT Klatt	U.S.A.	Text-to-Speech synthesizer
TI Fisher	U.S.A.	Text-to-Speech development system
MIT Searle et AL.	U.S.A./Canada	Speech analysis synthesis based on perception
BNR O'Shaughnessy	Canada	Text-to-speech
CNET Vaissiere et AL.	France	Prosody

ARCHITECTURES

<u>Institute</u>	<u>Country</u>	<u>Research</u>
Logica LTD (PECKMAN et AL., 1982)	England	Machines for computing DP matching in real-time
Purdue University (SIEGEL et AL.)	U.S.A.	SIMD machine for DP matching, complexity evaluation
Fairchild (LYON, 1982)	U.S.A.	Cochlear model
Brown University (SILVERMAN, 1982)	U.S.A.	Machine for computing DP matching with acoustic constraints
Concordia University (DE MORI, 1982)	Canada	Network architectures for lexical access
University California Berkeley (BRODERSON, 1982)	U.S.A.	VLSI DP matching
Carnegie Mellon Univ. (BISIANI, 1983)	U.S.A.	Architectures for problem solving
Bell Labs (WASTE et AL.)	U.S.A.	Systolic chip for DP matching
Verbex (McALLISTER)	U.S.A.	Systolic chip for stochastic decoding on a Markov model
NEC (Ishizuko et AL.)	Japan	Speech recognition processor
INTEL CHI FOON et AL.	U.S.A.	Speech recognition board

PHRASE RECOGNITION

<u>Institution</u>	<u>Country</u>	<u>Research</u>
Nippon Electric Co. (SAKOE, 1979)	Japan	Two levels DP
JSRU (BRIDLE et AL., 1982)	England	Two levels DP
Naval Res. Lab. Washington (SHORE & BURTON, 1982)	U.S.A.	Two levels DP
Carnegie Mellon Univ. (SMITH & ERMAN, 1981)	U.S.A.	Large lexicon in continuous speech
Kyoto Institute of Technology (NIIMI, 1979)	Japan	Spoken basic
NTT (SHIKANO KOHDA, 1978)	Japan	Automated travel infor- mation & reservation system
Bell Labs (LEVINSON, 1978)	U.S.A.	Automated travel infor- mation & reservation system
Auricle (WHITE, 1978)	U.S.A.	Viterbi algorithm
IBM Watson Center (BAHL et AL., 1981)	U.S.A.	Markov chains
University of Kiev (VINTSJUK, 1980)	URSS	Dynamic programming
University of Yamanashi (SHIGENAGA, 1979)	Japan	Spoken FORTRAN
Bell Labs (RABINER & LEVINSON, 1981)	U.S.A.	Use of syntactic constraints in DP matching
Concordia University (DE MORI, 1982)	Canada	Speaker-independent connected letters & digits

RECOGNITION OF PHONETIC FEATURES

<u>Institution</u>	<u>Country</u>	<u>Research</u>
BBN(Cambridge) (ROUCOS et AL., 1982)	U.S.A.	Use of dyphons
Hitachi (KOMATSU et AL., 1982)	Japan	Phoneme recognition
Toshiba (WATANABE et AL., 1982)	Japan	Syllabic sounds
University of Munich (RUSKE, 1982)	Germany	Syllabic sounds
MIT (ZUE et AL., 1982)	U.S.A.	Spectrogram reading
Electrotechnical Lab. Tokyo (NAKAJIMA et AL., 1982)	Japan	Phoneme recognition
Carnegie-Mellon Univ. (COLE et AL., 1982)	U.S.A.	Use of phonetic features
Academy of Sciences Moscow (KNIPPER, 1980)	URSS	Use of phonetic features
University of Lvov (GURA et AL., 1980)	URSS	Phoneme recognition
Tokyo University (FUJISAKI, 1982)	Japan	Plosive sounds recognition
Canberra University	Australia	Plosive sounds recognition
Concordia University (SUEN & SANTERRE, 1981)	Canada	Plosive sounds recognition
Concordia University (DE MORI, 1982)	Canada	Expert system for speech decoding
Centre National Edutes Telecommunications (MERCIER, 1983)	France	Expert system for speech decoding

Frame Instantiations by PPF

```

(LTH3
  ((LDEPDIP( 8, 5, 1)) result_of P_READ)
  (filled_by FR_STATE 1)

(FR_STATE 1
  (*(filled_by FR24)
  (( PS # = FALSE) NIL)
  (( LPEAK( 8, 13, 2)) result_of P_READ)
  (filled_by FR_STATE 2)

(FR_STATE 2
  (*(filled_by FR 8)
  (described_as (NI* ( 8- 5, 5, 7)))
  (described_as (VFC ( 7- 13, 1#)))
  (( PS 1 = TRUE)
  ((LDEPDIP( 13, 18, 3)) result_of P_READ)
  (filled_by FR_STATE 1)

(FR_STATE 1
  (*(filled_by FR24)
  (( PS # = FALSE) NIL)
  (( LPEAK( 18, 29, 4)) result_of P_READ)
  (filled_by FR_STATE 2)

(FR_STATE 2
  (*(filled_by FR 8)
  (described_as (NI* ( 13- 18, 18, 2#)))
  (described_as (VFC ( 2#- 29, 25)))
  (( PS 1 = TRUE)
  ((LDEPDIP( 29, 41, 5)) result_of P_READ)
  (filled_by FR_STATE 1)

(FR_STATE 1
  (*(filled_by FR24)
  (( PS # = FALSE) NIL)
  (( MPEAK( 41, 47, 6)) result_of P_READ)
  (filled_by FR_STATE 2)

(FR_STATE 2
  (*(filled_by FR14)
  (described_as (NI ( 41- 44)))
  (( PS 3 = TRUE)
  (( LPEAK( 47, 78, 7)) result_of P_READ)
  (filled_by FR_STATE 1)

(FR_STATE 1
  (*(filled_by FR 9)
  (described_as (VFC ( 47- 58, 52)))
  (( PS 1 = TRUE)
  ((SDEPDIP( 78, 72, 8)) result_of P_READ)
  (filled_by FR_STATE 1)

(FR_STATE 1
  (( PS # = FALSE) NIL)
  (( MPEAK( 72, 78, 9)) result_of P_READ)
  (filled_by FR_STATE 17)

(FR_STATE 17
  (*(filled_by FR14)
  (described_as (NI ( 72- 75, 75)))
  (described_as (VF* ( 76- 78)))
  (( PS 3 = FALSE) NIL)
  ((SHIGHDIP( 78, 79, 1#)) result_of P_READ)
  (filled_by FR_STATE 18)

(FR_STATE 18
  (*(filled_by FR 4)
  (described_as (SON* ( 78, 79)))
  (( PS # = FALSE) NIL)
  (( MNS( 82, 87, 11)) result_of P_READ)
  (filled_by FR_STATE 2#)

(FR_STATE 2#
  (*(filled_by FR 3)
  (described_as (NC ( 82, 87)))
  (( PS # = FALSE) NIL)
  ((SHIGHDIP( 88, 91, 12)) result_of P_READ)
  (filled_by FR_STATE 29)

(FR_STATE 29
  (( PS # = FALSE) NIL)
  (( LPEAK( 91, 1#1, 13)) result_of P_READ)
  (filled_by FR_STATE 25)

(FR_STATE 25
  (*(filled_by FR 2)
  (described_as (VFC* ( 91-1#1, 95)))
  (( PS 1 = TRUE)
  (( SMEDDIP(1#1, 1#2, 14)) result_of P_READ)
  (filled_by FR_STATE 1)

(FR_STATE 1
  (*(filled_by FR 5)
  (described_as (SON*(1#1, 1#2)))
  (( PS # = FALSE) NIL)
  (( LPEAK(1#2, 114, 15)) result_of P_READ)
  (filled_by FR_STATE 31)

(FR_STATE 31
  (*(filled_by FR 5)
  (described_as (NC (1#2, 1#7)))
  (described_as (VFC* (1#8-114, 111)))
  (( PS 3 = TRUE)
  ((SDEPDIP(114, 117, 16)) result_of P_READ)
  (filled_by FR_STATE 1)

(FR_STATE 1
  (( PS # = FALSE) NIL)
  (( LPEAK(117, 128, 17)) result_of P_READ)
  (filled_by FR_STATE 17)

(FR_STATE 17
  (*(filled_by FR 5)
  (described_as (VFC (117-128, 123)))
  (( PS 3 = FALSE) NIL)
  (( SMEDDIP(128, 129, 18)) result_of P_READ)
  (filled_by FR_STATE 19)

(FR_STATE 19
  (*(filled_by FR 4)
  (described_as (SON*(128, 129)))
  (( PS # = FALSE) NIL)
  (( LPEAK(129, 148, 19)) result_of P_READ)
  (filled_by FR_STATE 2#)

(FR_STATE 2#
  (*(filled_by FR 2)
  (( PS 1 = TRUE)
  ((LDEPDIP(148, 148, 2#)) result_of P_READ)
  (filled_by FR_STATE 1)

(FR_STATE 1
  (*(filled_by FR24)

```


APPENDIX B-II

Frame Instantiations by SPE

(FRI14
 (INT14 (result of P-INT14(INPUT)))
 (LPCVAL (result of P-LPC(M-INT14)))
 (R14 (result of F-TENG(M-INT14)))

(when P14-P
 (filled by P-VOCALIC))

(when (AND(NOT(P14-P),OR(NOT(HRM-P(M-INT14),
 P141-P))))
 (described as NI(INT141)))

(when POSSVOC-P)
 (described as VF* (INT142)))

(when (AND(NOT(P14-P),P142-P))
 (described as NC(INT14)))

(when (AND(NOT(P14-P),NOT(P142-P))
 (described as SONV(INT14)))

(P-VOCALIC

(when VOCP1-P (LPCVAL
 (filled by P-DESCR))

(when VOCP2-P (LPCVAL
 (described as NC(INT14))))

P-LPC is the procedure evaluating the LPC at the middle of the interval INT14.

P-DESCR is the procedure to describe the vowel according to the frequency energy spectrum.

F-TENG evaluates the difference the total energy from its derivative.

P14-P is true if it is a long peak with low zero crossings and high R14 or the peak is followed by a deep-dip or the peak is preceded by a long medium or high dip.

P141-P is true if it is not a long peak and is preceded by a deep dip.

P142-P is true if the zero-crossing in the peak is higher than 70.

POSSVOC-P is true when the peak is preceded by a NS and the energy amplitude of the first three formants exceed 7db with the finest formant lies below 500 Hz.

INT141 is the first three time frames in INT14.

INT142 is the first four time frames in INT14.

HRM-P is true when the difference of the total energy from its derivatives is large while the derivative considerably high.

VOCP1-P is true when the frequency-energy spectrum indicates vocalic properties.

VOCP2-P is true when the first formant is degenerated and the second formant lies in the range of 1000 Hz and 1400 Hz.

114
APPENDIX C

Test Sentence "A Good Turn Deserves Another"

