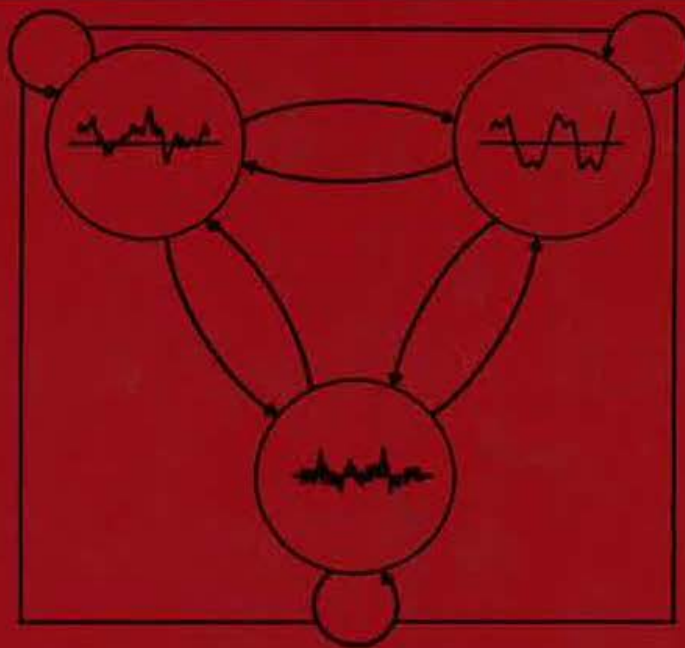


FUNDAMENTALS OF SPEECH RECOGNITION



LAWRENCE RABINER
BIING-HWANG JUANG

Petitioner has added romanettes to pages (i) - (ii). Otherwise, it leaves the original page numbering.

MICROSOFT CORP.
EXHIBIT 1018

Library of Congress Cataloging-in-Publication Data

Rabiner, Lawrence R., 1943-

Fundamentals of speech recognition / Lawrence Rabiner, Bing-Hwang Juang.

p. cm.

Includes bibliographical references and index.

ISBN 0-13-015157-2

1. Automatic speech recognition. 2. Speech processing systems.

I. Juang, B. H. (Bing-Hwang) II. Title.

TK7895.S65R33 1993

006.4'54—dc20

92-34093

CIP

Editorial production

Cover Designer: *Ben Santora*

and interior design: *bookworks*

Acquisitions Editor: *Karen Gettman*

Manufacturing Buyer: *Mary Elizabeth McCartney*

©1993 by AT&T. All rights reserved.



Published by PTR Prentice-Hall, Inc.

A Simon & Schuster Company

Englewood Cliffs, New Jersey 07632

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact:

Corporate Sales Department

PTR Prentice Hall

113 Sylvan Avenue

Englewood Cliffs, NJ 07632

Phone: 201-592-2863

FAX: 201-592-2249

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-015157-2

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

CONTENTS

LIST OF FIGURES	xiii
LIST OF TABLES	xxix
PREFACE	xxxi
1 FUNDAMENTALS OF SPEECH RECOGNITION	1
1.1 Introduction	1
1.2 The Paradigm for Speech Recognition	3
1.3 Outline	3
1.4 A Brief History of Speech-Recognition Research	6
2 THE SPEECH SIGNAL: PRODUCTION, PERCEPTION, AND ACOUSTIC-PHONETIC CHARACTERIZATION	11
2.1 Introduction	11
2.1.1 The Process of Speech Production and Perception in Human Beings	11
2.2 The Speech-Production Process	14
2.3 Representing Speech in the Time and Frequency Domains	17
2.4 Speech Sounds and Features	20
	vii

2.4.1	The Vowels	21
2.4.2	Diphthongs	28
2.4.3	Semivowels	29
2.4.4	Nasal Consonants	30
2.4.5	Unvoiced Fricatives	31
2.4.6	Voiced Fricatives	32
2.4.7	Voiced and Unvoiced Stops	33
2.4.8	Review Exercises	37
2.5	Approaches to Automatic Speech Recognition by Machine	42
2.5.1	Acoustic-Phonetic Approach to Speech Recognition	45
2.5.2	Statistical Pattern-Recognition Approach to Speech Recognition	51
2.5.3	Artificial Intelligence (AI) Approaches to Speech Recognition	52
2.5.4	Neural Networks and Their Application to Speech Recognition	54
2.6	Summary	65
3	SIGNAL PROCESSING AND ANALYSIS METHODS FOR SPEECH RECOGNITION	69
3.1	Introduction	69
3.1.1	Spectral Analysis Models	70
3.2	The Bank-of-Filters Front-End Processor	73
3.2.1	Types of Filter Bank Used for Speech Recognition	77
3.2.2	Implementations of Filter Banks	80
3.2.3	Summary of Considerations for Speech-Recognition Filter Banks	92
3.2.4	Practical Examples of Speech-Recognition Filter Banks	93
3.2.5	Generalizations of Filter-Bank Analyzer	95
3.3	Linear Predictive Coding Model for Speech Recognition	97
3.3.1	The LPC Model	100
3.3.2	LPC Analysis Equations	101
3.3.3	The Autocorrelation Method	103
3.3.4	The Covariance Method	106
3.3.5	Review Exercise	107
3.3.6	Examples of LPC Analysis	108
3.3.7	LPC Processor for Speech Recognition	112
3.3.8	Review Exercises	117
3.3.9	Typical LPC Analysis Parameters	121
3.4	Vector Quantization	122
3.4.1	Elements of a Vector Quantization Implementation	123
3.4.2	The VQ Training Set	124
3.4.3	The Similarity or Distance Measure	125
3.4.4	Clustering the Training Vectors	125
3.4.5	Vector Classification Procedure	128
3.4.6	Comparison of Vector and Scalar Quantizers	129

3.4.7	Extensions of Vector Quantization	129
3.4.8	Summary of the VQ Method	131
3.5	Auditory-Based Spectral Analysis Models	132
3.5.1	The EIH Model	134
3.6	Summary	139

4 PATTERN-COMPARISON TECHNIQUES 141

4.1	Introduction	141
4.2	Speech (Endpoint) Detection	143
4.3	Distortion Measures—Mathematical Considerations	149
4.4	Distortion Measures—Perceptual Considerations	150
4.5	Spectral-Distortion Measures	154
4.5.1	Log Spectral Distance	158
4.5.2	Cepstral Distances	163
4.5.3	Weighted Cepstral Distances and Liftering	166
4.5.4	Likelihood Distortions	171
4.5.5	Variations of Likelihood Distortions	177
4.5.6	Spectral Distortion Using a Warped Frequency Scale	183
4.5.7	Alternative Spectral Representations and Distortion Measures	190
4.5.8	Summary of Distortion Measures—Computational Considerations	193
4.6	Incorporation of Spectral Dynamic Features into the Distortion Measure	194
4.7	Time Alignment and Normalization	200
4.7.1	Dynamic Programming—Basic Considerations	204
4.7.2	Time-Normalization Constraints	208
4.7.3	Dynamic Time-Warping Solution	221
4.7.4	Other Considerations in Dynamic Time Warping	229
4.7.5	Multiple Time-Alignment Paths	232
4.8	Summary	238

5 SPEECH RECOGNITION SYSTEM DESIGN AND IMPLEMENTATION ISSUES 242

5.1	Introduction	242
5.2	Application of Source-Coding Techniques to Recognition	244
5.2.1	Vector Quantization and Pattern Comparison Without Time Alignment	244
5.2.2	Centroid Computation for VQ Codebook Design	246
5.2.3	Vector Quantizers with Memory	254
5.2.4	Segmental Vector Quantization	256
5.2.5	Use of a Vector Quantizer as a Recognition Preprocessor	257
5.2.6	Vector Quantization for Efficient Pattern Matching	263
5.3	Template Training Methods	264
5.3.1	Casual Training	265

5.3.2	Robust Training	266
5.3.3	Clustering	267
5.4	Performance Analysis and Recognition Enhancements	274
5.4.1	Choice of Distortion Measures	274
5.4.2	Choice of Clustering Methods and kNN Decision Rule	277
5.4.3	Incorporation of Energy Information	280
5.4.4	Effects of Signal Analysis Parameters	282
5.4.5	Performance of Isolated Word-Recognition Systems	284
5.5	Template Adaptation to New Talkers	285
5.5.1	Spectral Transformation	286
5.5.2	Hierarchical Spectral Clustering	288
5.6	Discriminative Methods in Speech Recognition	291
5.6.1	Determination of Word Equivalence Classes	294
5.6.2	Discriminative Weighting Functions	297
5.6.3	Discriminative Training for Minimum Recognition Error	302
5.7	Speech Recognition in Adverse Environments	305
5.7.1	Adverse Conditions in Speech Recognition	306
5.7.2	Dealing with Adverse Conditions	309
5.8	Summary	317

6 THEORY AND IMPLEMENTATION OF HIDDEN MARKOV MODELS 321

6.1	Introduction	321
6.2	Discrete-Time Markov Processes	322
6.3	Extensions to Hidden Markov Models	325
6.3.1	Coin-Toss Models	326
6.3.2	The Urn-and-Ball Model	328
6.3.3	Elements of an HMM	329
6.3.4	HMM Generator of Observations	330
6.4	The Three Basic Problems for HMMs	333
6.4.1	Solution to Problem 1—Probability Evaluation	334
6.4.2	Solution to Problem 2—"Optimal" State Sequence	337
6.4.3	Solution to Problem 3—Parameter Estimation	342
6.4.4	Notes on the Reestimation Procedure	347
6.5	Types of HMMs	348
6.6	Continuous Observation Densities in HMMs	350
6.7	Autoregressive HMMs	352
6.8	Variants on HMM Structures—Null Transitions and Tied States	356
6.9	Inclusion of Explicit State Duration Density in HMMs	358
6.10	Optimization Criterion—ML, MMI, and MDI	362
6.11	Comparisons of HMMs	364
6.12	Implementation Issues for HMMs	365
6.12.1	Scaling	365
6.12.2	Multiple Observation Sequences	369
6.12.3	Initial Estimates of HMM Parameters	370

6.12.4	Effects of Insufficient Training Data	370
6.12.5	Choice of Model	371
6.13	Improving the Effectiveness of Model Estimates	372
6.13.1	Deleted Interpolation	372
6.13.2	Bayesian Adaptation	373
6.13.3	Corrective Training	376
6.14	Model Clustering and Splitting	377
6.15	HMM System for Isolated Word Recognition	378
6.15.1	Choice of Model Parameters	379
6.15.2	Segmental K-Means Segmentation into States	382
6.15.3	Incorporation of State Duration into the HMM	384
6.15.4	HMM Isolated-Digit Performance	385
6.16	Summary	386
7	SPEECH RECOGNITION BASED ON CONNECTED WORD MODELS	390
7.1	Introduction	390
7.2	General Notation for the Connected Word-Recognition Problem	393
7.3	The Two-Level Dynamic Programming (Two-Level DP) Algorithm	395
7.3.1	Computation of the Two-Level DP Algorithm	399
7.4	The Level Building (LB) Algorithm	400
7.4.1	Mathematics of the Level Building Algorithm	401
7.4.2	Multiple Level Considerations	405
7.4.3	Computation of the Level Building Algorithm	407
7.4.4	Implementation Aspects of Level Building	410
7.4.5	Integration of a Grammar Network	414
7.4.6	Examples of LB Computation of Digit Strings	416
7.5	The One-Pass (One-State) Algorithm	416
7.6	Multiple Candidate Strings	420
7.7	Summary of Connected Word Recognition Algorithms	423
7.8	Grammar Networks for Connected Digit Recognition	425
7.9	Segmental K-Means Training Procedure	427
7.10	Connected Digit Recognition Implementation	428
7.10.1	HMM-Based System for Connected Digit Recognition	429
7.10.2	Performance Evaluation on Connected Digit Strings	430
7.11	Summary	432
8	LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION	434
8.1	Introduction	434
8.2	Subword Speech Units	435
8.3	Subword Unit Models Based on HMMs	439
8.4	Training of Subword Units	441

8.5	Language Models for Large Vocabulary Speech Recognition	447
8.6	Statistical Language Modeling	448
8.7	Perplexity of the Language Model	449
8.8	Overall Recognition System Based on Subword Units	450
8.8.1	Control of Word Insertion/Word Deletion Rate	454
8.8.2	Task Semantics	454
8.8.3	System Performance on the Resource Management Task	454
8.9	Context-Dependent Subword Units	458
8.9.1	Creation of Context-Dependent Diphones and Triphones	460
8.9.2	Using Interword Training to Create CD Units	461
8.9.3	Smoothing and Interpolation of CD PLU Models	462
8.9.4	Smoothing and Interpolation of Continuous Densities	464
8.9.5	Implementation Issues Using CD Units	464
8.9.6	Recognition Results Using CD Units	467
8.9.7	Position Dependent Units	469
8.9.8	Unit Splitting and Clustering	470
8.9.9	Other Factors for Creating Additional Subword Units	475
8.9.10	Acoustic Segment Units	476
8.10	Creation of Vocabulary-Independent Units	477
8.11	Semantic Postprocessor for Recognition	478
8.12	Summary	478

9 TASK ORIENTED APPLICATIONS OF AUTOMATIC SPEECH RECOGNITION 482

9.1	Introduction	482
9.2	Speech-Recognizer Performance Scores	484
9.3	Characteristics of Speech-Recognition Applications	485
9.3.1	Methods of Handling Recognition Errors	486
9.4	Broad Classes of Speech-Recognition Applications	487
9.5	Command-and-Control Applications	488
9.5.1	Voice Repertory Dialer	489
9.5.2	Automated Call-Type Recognition	490
9.5.3	Call Distribution by Voice Commands	491
9.5.4	Directory Listing Retrieval	491
9.5.5	Credit Card Sales Validation	492
9.6	Projections for Speech Recognition	493

INDEX 497

LIST OF FIGURES

1.1	General block diagram of a task-oriented speech-recognition system.	3
2.1	Schematic diagram of speech-production/speech-perception process (after Flanagan [unpublished]).	12
2.2	Alternative view of speech-production/speech-perception process (after Rabiner and Levinson [1]).	13
2.3	Mid-sagittal plane X-ray of the human vocal apparatus (after Flanagan et al. [2]).	15
2.4	Schematic view of the human vocal mechanism (after Flanagan [3]).	16
2.5	Glottal volume velocity and resulting sound pressure at the start of a voiced sound (after Ishizaka and Flanagan [4]).	16
2.6	Schematic representation of the complete physiological mechanism of speech production (after Flanagan [3]).	17
2.7	Waveform plot of the beginning of the utterance "It's time."	18
2.8	Wideband and narrowband spectrograms and speech amplitude for the utterance "Every salt breeze comes from the sea."	19

THE SPEECH SIGNAL: PRODUCTION, PERCEPTION, AND ACOUSTIC-PHONETIC CHARACTERIZATION

2.1 INTRODUCTION

In this chapter we discuss the mechanics of producing and perceiving speech in human beings, and we show how an understanding of these processes leads naturally to several different approaches to speech recognition by machine. We begin by showing how the different classes of speech sounds, or phonetics, can each be characterized in terms of broad acoustic features whose properties are relatively invariant across words and speakers. The ideas of acoustic-phonetic characterization of sounds lead naturally to straightforward implementation of a speech-recognition algorithm based on sequential detection of sounds and sound classes. The strengths and weaknesses of such an approach are discussed. An alternative approach to speech recognition is to use standard pattern-recognition techniques in a framework in which all speech knowledge is “learned” via a training phase. We show that such a “blind” approach has some natural advantages for a wide range of speech-recognition systems. Finally we show how aspects of both the acoustic-phonetic approach and the pattern-recognition approach can be integrated into a hybrid method that includes techniques from artificial intelligence as well as neural network methods.

2.1.1 The Process of Speech Production and Perception in Human Beings

Figure 2.1 shows a schematic diagram of the speech-production/speech-perception process in human beings. The production (speech-generation) process begins when the talker

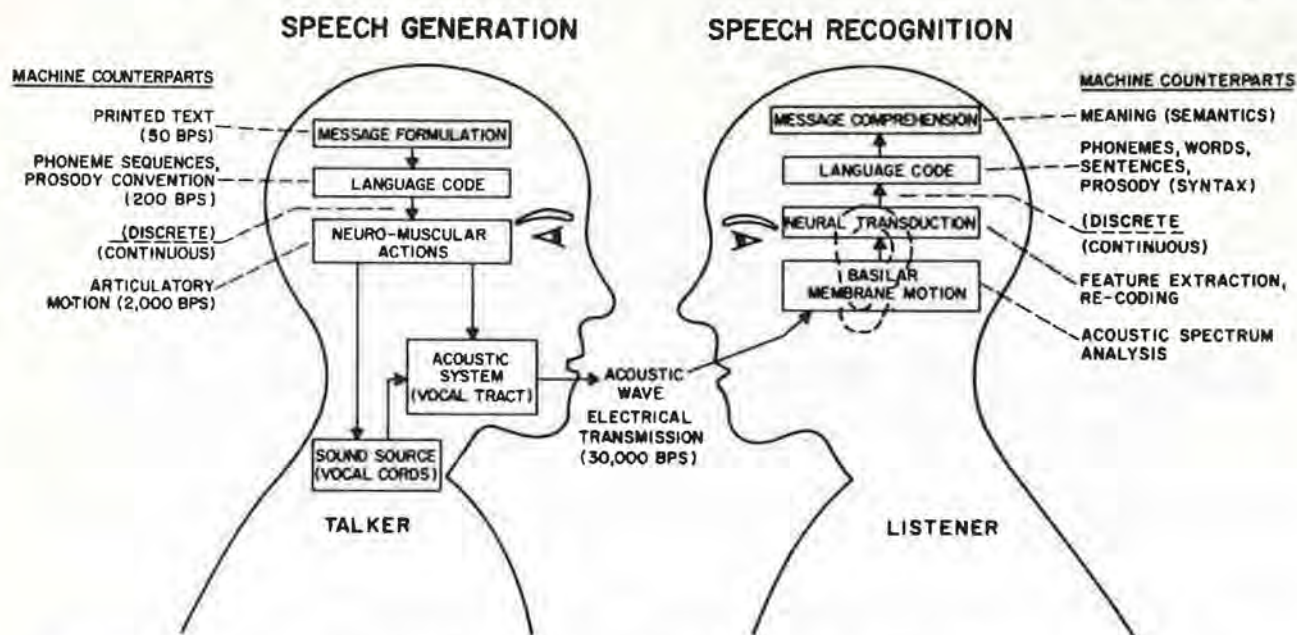


Figure 2.1 Schematic diagram of speech-production/speech-perception process (after Flanagan [unpublished]).

formulates a message (in his mind) that he wants to transmit to the listener via speech. The machine counterpart to the process of message formulation is the creation of printed text expressing the words of the message. The next step in the process is the conversion of the message into a language code. This roughly corresponds to converting the printed text of the message into a set of phoneme sequences corresponding to the sounds that make up the words, along with prosody markers denoting duration of sounds, loudness of sounds, and pitch accent associated with the sounds. Once the language code is chosen, the talker must execute a series of neuromuscular commands to cause the vocal cords to vibrate when appropriate and to shape the vocal tract such that the proper sequence of speech sounds is created and spoken by the talker, thereby producing an acoustic signal as the final output. The neuromuscular commands must simultaneously control all aspects of articulatory motion including control of the lips, jaw, tongue, and velum (a “trapdoor” controlling the acoustic flow to the nasal mechanism).

Once the speech signal is generated and propagated to the listener, the speech-perception (or speech-recognition) process begins. First the listener processes the acoustic signal along the basilar membrane in the inner ear, which provides a running spectrum analysis of the incoming signal. A neural transduction process converts the spectral signal at the output of the basilar membrane into activity signals on the auditory nerve, corresponding roughly to a feature extraction process. In a manner that is not well understood, the neural activity along the auditory nerve is converted into a language code at the higher centers of processing within the brain, and finally message comprehension (understanding of meaning) is achieved.

A slightly different view of the speech-production/speech-perception process is shown in Figure 2.2. Here we see the steps in the process laid out along a line corresponding to the basic information rate of the signal (or control) at various stages of the

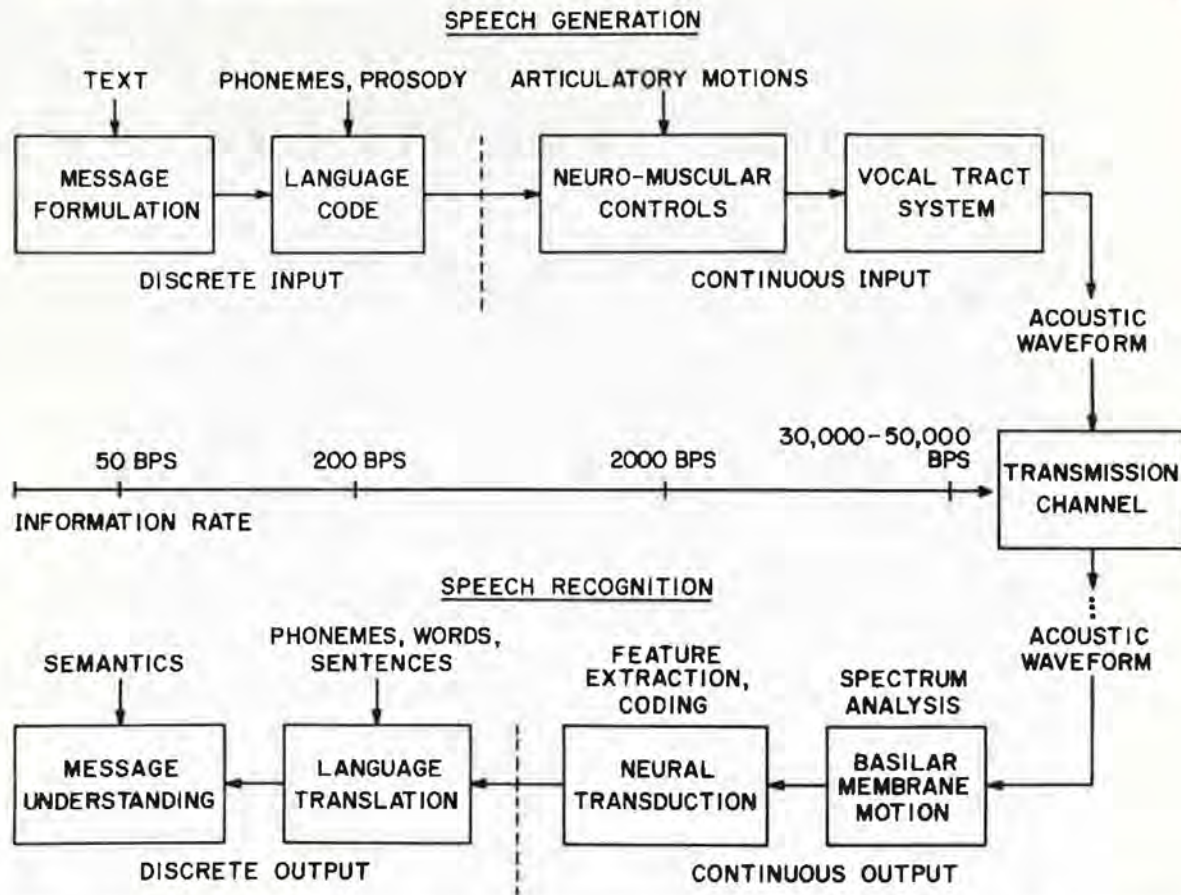


Figure 2.2 Alternative view of speech-production/speech-perception process (after Rabiner and Levinson [1]).

process. The discrete symbol information rate in the raw message text is rather low (about 50 bps [bits per second] corresponding to about 8 sounds per second, where each sound is one of about 50 distinct symbols). After the language code conversion, with the inclusion of prosody information, the information rate rises to about 200 bps. Somewhere in the next stage the representation of the information in the signal (or the control) becomes continuous with an equivalent rate of about 2000 bps at the neuromuscular control level, and about 30,000–50,000 bps at the acoustic signal level.

A transmission channel is shown in Figure 2.2 [1], indicating that any of several well-known coding techniques could be used to transmit the acoustic waveform from the talker to the listener. The steps in the speech-perception mechanism can also be interpreted in terms of information rate in the signal or its control and follows the inverse pattern of the production process. Thus the continuous information rate at the basilar membrane is in the range of 30,000–50,000 bps, while at the neural transduction stage it is about 2000 bps. The higher-level processing within the brain converts the neural signals to a discrete representation, which ultimately is decoded into a low-bit-rate message.

To illustrate, in a trivial way, how the speech-production/speech-perception process works, consider that the speaker has a goal of finding out whether his office mate has eaten his lunch yet. To express this thought, the speaker formulates the message “Did you eat

yet?” In the process of converting the message to a language code, the text becomes a phonetic sequence of sounds of the form /dI d yu it yet?/, in which each word is expressed as a sequence of phonemes constituting the ideal pronunciation of the sounds of the word (as spoken in isolation) within the spoken language. However, because the words are not spoken in isolation, and a physical mechanism is used to produce the sounds (the human vocal tract system), and because physical systems obey continuity and smoothness constraints, by the time the message is spoken the sounds become more like the phonetic string /dI jə it jət?/. The final d in dId is dropped, the word *you* becomes converted to a word that sounds a lot like “juh,” and finally the word *yet* is pronounced as “jet.” Remarkably, through the speech-perception process, human beings are usually able to decode this highly stylized version of the text into the correct string; sadly, however, this remains a most difficult task for almost all speech-recognition machines.

2.2 THE SPEECH-PRODUCTION PROCESS

Figure 2.3 shows a mid-sagittal plane (longitudinal cross-section) X-ray of the human vocal apparatus [2]. The *vocal tract*, outlined by the dotted lines in Figure 2.3, begins at the opening of the vocal cords, or *glottis*, and ends at the lips. The vocal tract consists of the *pharynx* (the connection from the esophagus to the mouth) and the mouth, or *oral cavity*. In the average male, the total length of the vocal tract is about 17 cm. The cross-sectional area of the vocal tract, determined by the positions of the tongue, lips, jaw, and velum, varies from zero (complete closure) to about 20 cm². The *nasal tract* begins at the velum and ends at the nostrils. When the *velum*, (a trapdoor-like mechanism at the back of the mouth cavity) is lowered, the nasal tract is acoustically coupled to the vocal tract to produce the nasal sounds of speech.

A schematic diagram of the human vocal mechanism is shown in Figure 2.4 [3]. Air enters the lungs via the normal breathing mechanism. As air is expelled from the lungs through the *trachea* (or windpipe), the tensed vocal cords within the *larynx* are caused to vibrate (in the mode of a relaxation oscillator) by the air flow. The air flow is chopped into quasi-periodic pulses which are then modulated in frequency in passing through the *pharynx* (the throat cavity), the mouth cavity, and possibly the nasal cavity. Depending on the positions of the various articulators (i.e., jaw, tongue, velum, lips, mouth), different sounds are produced.

Figure 2.5 shows plots of the glottal air flow (volume velocity waveform) and the resulting sound pressure at the mouth for a typical vowel sound [4]. The glottal waveform shows a gradual build-up to a quasi-periodic pulse train of air, taking about 15 msec to reach steady state. This build-up is also reflected in the acoustic waveform shown at the bottom of the figure.

A simplified representation of the complete physiological mechanism for creating speech is shown in Figure 2.6 [3]. The lungs and the associated muscles act as the source of air for exciting the vocal mechanism. The muscle force pushes air out of the lungs (shown schematically as a piston pushing up within a cylinder) and through the bronchi and trachea. When the vocal cords are tensed, the air flow causes them to vibrate, producing

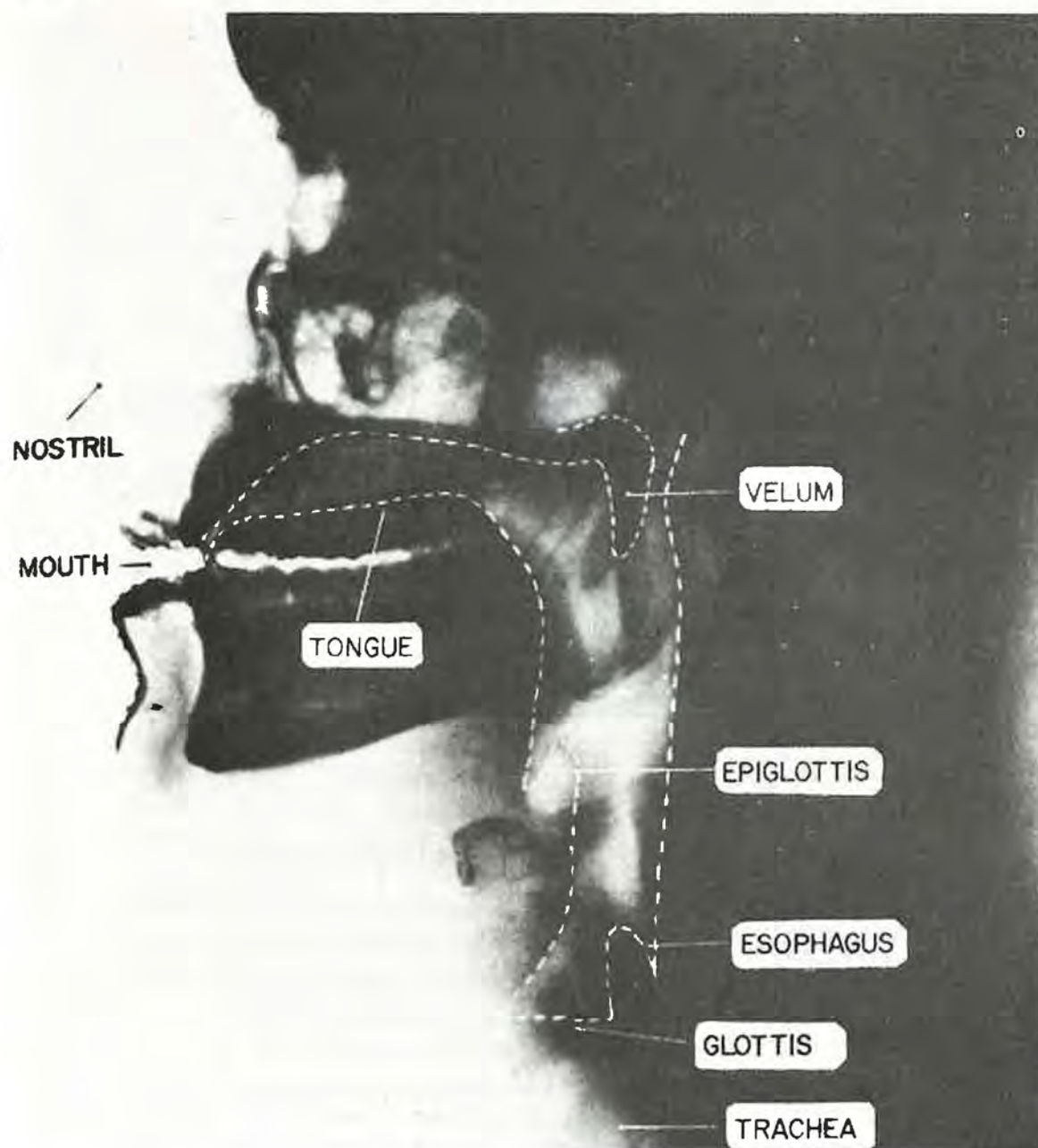


Figure 2.3 Mid-sagittal plane X-ray of the human vocal apparatus (after Flanagan et al. [2]).

so-called voiced speech sounds. When the vocal cords are relaxed, in order to produce a sound, the air flow either must pass through a constriction in the vocal tract and thereby become turbulent, producing so-called unvoiced sounds, or it can build up pressure behind a point of total closure within the vocal tract, and when the closure is opened, the pressure is suddenly and abruptly released, causing a brief transient sound.

Speech is produced as a sequence of sounds. Hence the state of the vocal cords, as well as the positions, shapes, and sizes of the various articulators, changes over time to reflect the sound being produced. The manner in which different sounds are created will be described later in this chapter. First we divert to a brief discussion of the speech waveform

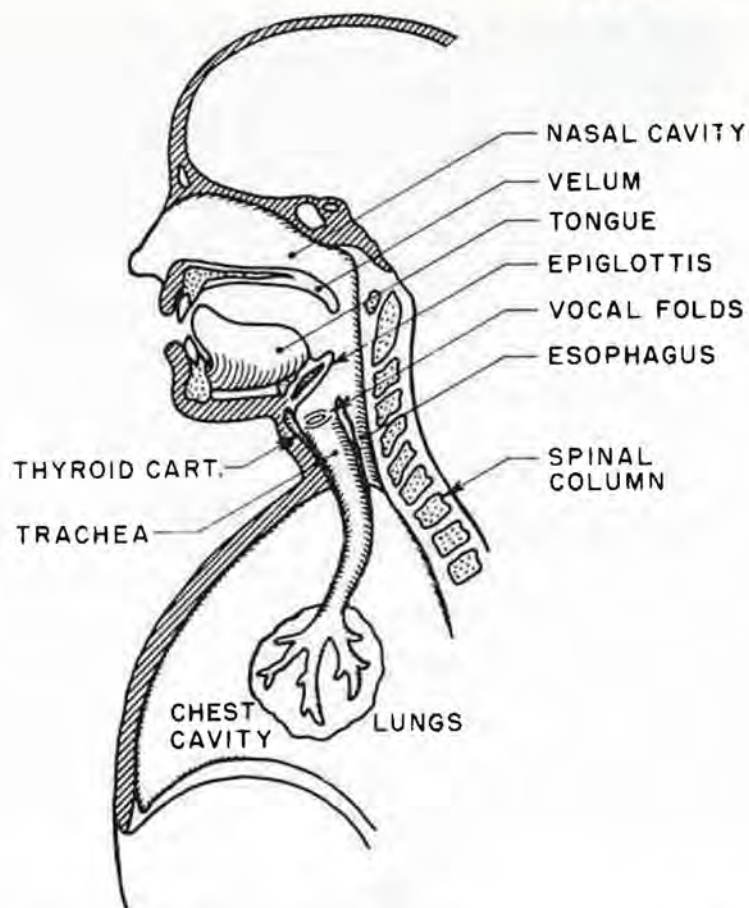


Figure 2.4 Schematic view of the human vocal mechanism (after Flanagan [3]).

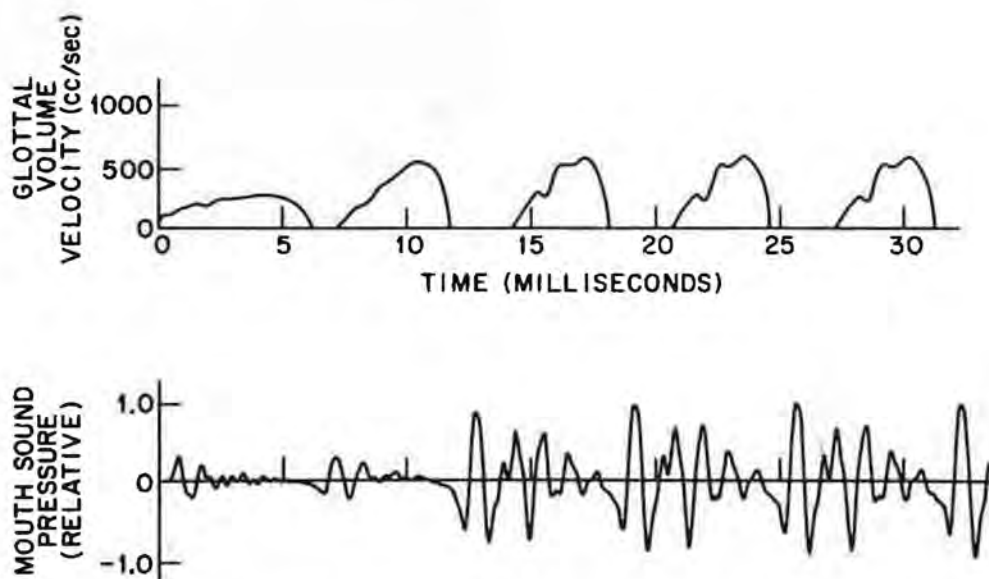


Figure 2.5 Glottal volume velocity and resulting sound pressure at the start of a voiced sound (after Ishizaka and Flanagan [4]).

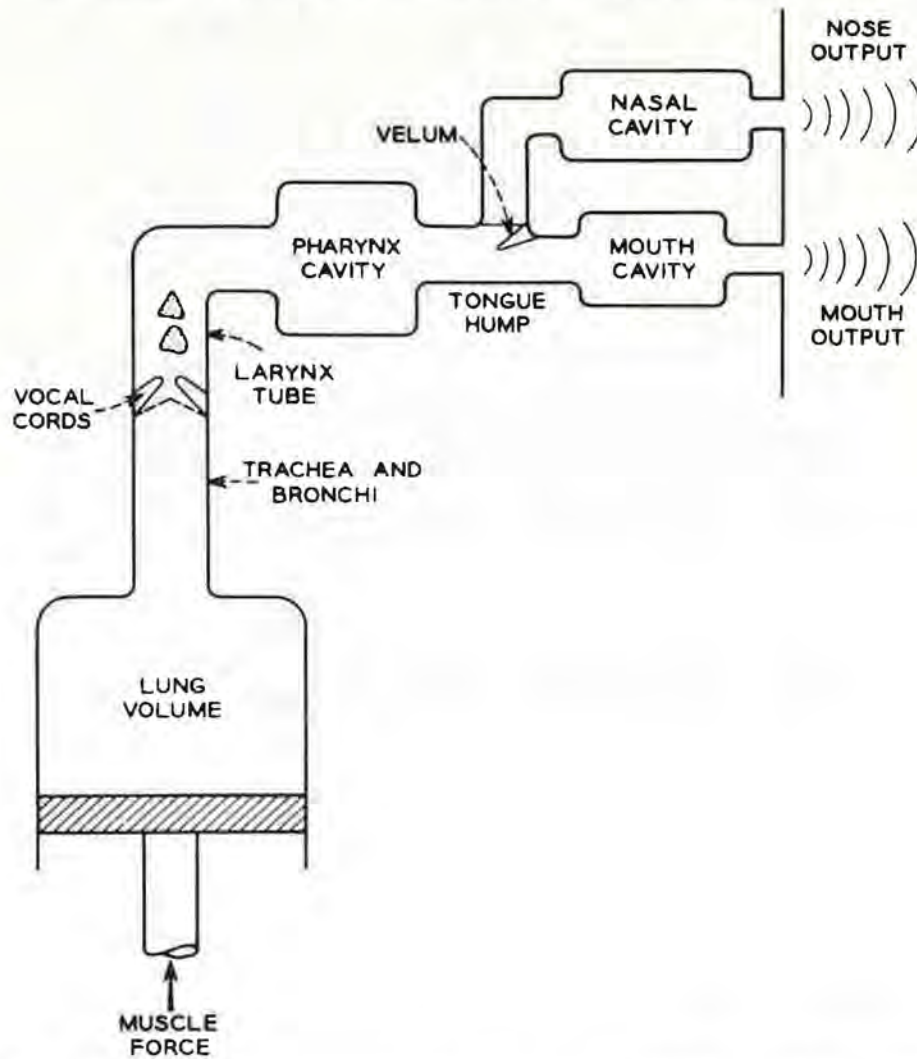


Figure 2.6 Schematic representation of the complete physiological mechanism of speech production (after Flanagan [3]).

and its spectral representation.

2.3 REPRESENTING SPEECH IN THE TIME AND FREQUENCY DOMAINS

The speech signal is a slowly time varying signal in the sense that, when examined over a sufficiently short period of time (between 5 and 100 msec), its characteristics are fairly stationary; however, over long periods of time (on the order of 1/5 seconds or more) the signal characteristics change to reflect the different speech sounds being spoken. An illustration of this effect is given in Figure 2.7, which shows the time waveform corresponding to the initial sounds in the phrase, "It's time . . ." as spoken by a male speaker. Each line of the waveform corresponds to 100 msec (1/10 second) of signal; hence the entire plot encompasses about 0.5 sec.

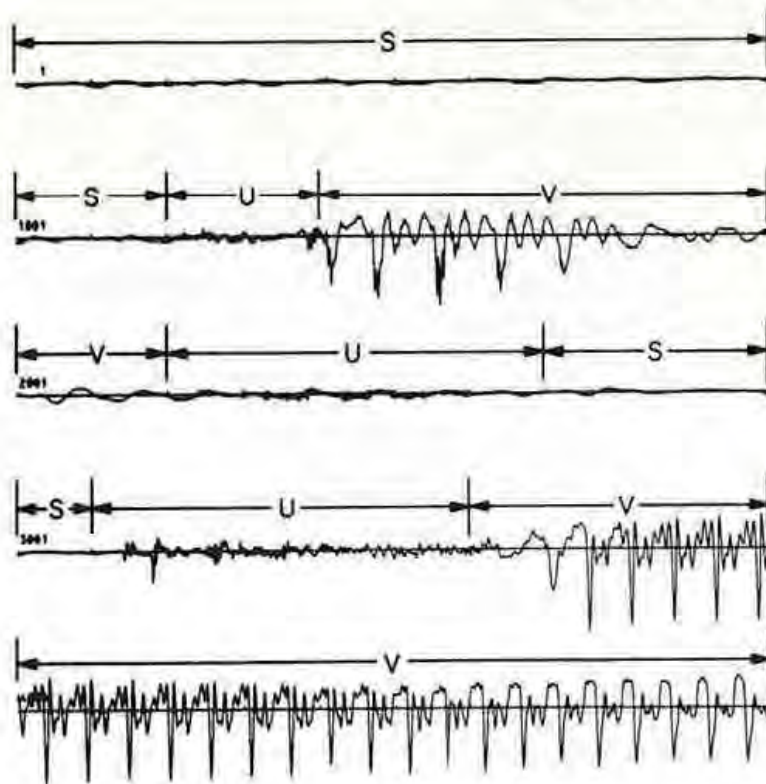


Figure 2.7 Waveform plot of the beginning of the utterance "It's time."

The slowly time varying nature of the signal can be seen by contrasting the first 100 msec of the waveform (the first line), which corresponds to background silence and is therefore low in amplitude, to the next 100 msec of the waveform (the second line), which first shows a small increase in level, and then a sharp increase in level and a gross change in waveform shape and regularity (it becomes almost periodic).

There are several ways of classifying (labeling) events in speech. Perhaps the simplest and most straightforward is via the state of the speech-production source—the vocal cords. It is accepted convention to use a three-state representation in which the states are (1) silence (S), where no speech is produced; (2) unvoiced (U), in which the vocal cords are not vibrating, so the resulting speech waveform is aperiodic or random in nature; and (3) voiced (V), in which the vocal cords are tensed and therefore vibrate periodically when air flows from the lungs, so the resulting speech waveform is quasi-periodic. The result of applying this type of classification to the waveform of Figure 2.7 is shown in the figure. Initially, before speaking begins, the waveform is classified as silence (S). A brief period of unvoiced (U) sound (whisper or aspiration) is seen prior to the voicing (V) corresponding to the initial vowel in the word *It's*. Following the voicing region, there is a brief, unvoiced aspiration (devoicing of the vowel), followed by a silence region (prior to the /t/ in *It's*), and then a relatively long, unvoiced (U) region corresponding to the /t/ release, followed by the /s/, followed by the /t/ in *time*. Finally there is a long voicing (V) region corresponding to the diphthong /a^y/ in *time*.

It should be clear that the segmentation of the waveform into well-defined regions of

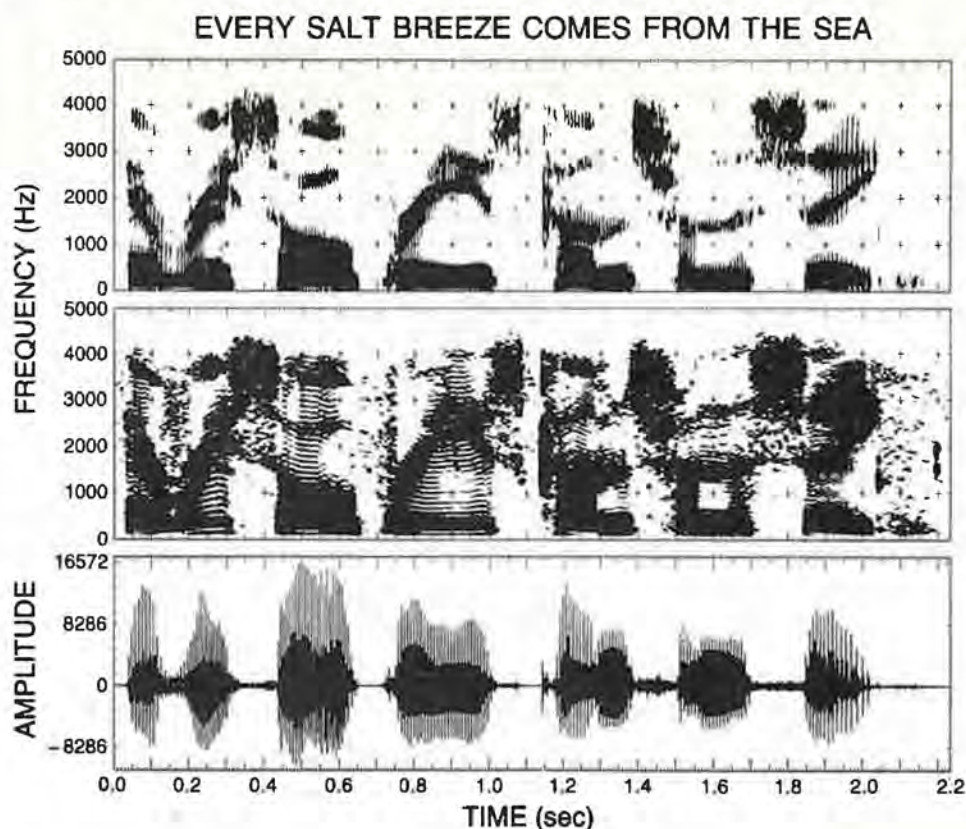


Figure 2.8 Wideband and narrowband spectrograms and speech amplitude for the utterance “Every salt breeze comes from the sea.”

silence, unvoiced, and voiced signals is not exact; it is often difficult to distinguish a weak, unvoiced sound (like /f/ or /th/) from silence, or a weak, voiced sound (like /v/ or /m/) from unvoiced sounds or even silence. However, it is usually not critical to segment the signal to a precision much less than several milliseconds; hence, small errors in boundary locations usually have no consequence for most applications.

An alternative way of characterizing the speech signal and representing the information associated with the sounds is via a spectral representation. Perhaps the most popular representation of this type is the sound spectrogram in which a three-dimensional representation of the speech intensity, in different frequency bands, over time is portrayed. An example of this type of speech representation is given in Figure 2.8, which shows a *wideband spectrogram* in the first panel, a *narrowband spectrogram* in the second panel, and a waveform amplitude plot in the third panel, of a spoken version of the utterance “Every salt breeze comes from the sea” by a male speaker. The wideband spectrogram corresponds to performing a spectral analysis on 15-msec sections of waveform using a broad analysis filter (125 Hz bandwidth) with the analysis advancing in intervals of 1 msec. The spectral intensity at each point in time is indicated by the intensity (darkness) of the plot at a particular analysis frequency. Because of the relatively broad bandwidth of the analysis filters, hence the relatively short duration of the analysis window, the spectral envelope of individual periods of the speech waveform during voiced sections are resolved

and are seen as vertical striations in the spectrogram.

The narrowband spectrogram (shown in the second panel of Figure 2.8) corresponds to performing a spectral analysis on 50-msec sections of waveform using a narrow analysis filter (40 Hz bandwidth), with the analysis again advancing in intervals of 1 msec. Because of the relatively narrow bandwidth of the analysis filters, individual spectral harmonics corresponding to the pitch of the speech waveform, during voiced regions, are resolved and are seen as almost-horizontal lines in the spectrogram. During periods of unvoiced speech, we see primarily high-frequency energy in the spectrograms; during periods of silence we essentially see no spectral activity (because of the reduced signal level).

A third way of representing the time-varying signal characteristics of speech is via a parameterization of the spectral activity based on the model of speech production. Because the human vocal tract is essentially a tube, or concatenation of tubes, of varying cross-sectional area that is excited either at one end (by the vocal cord puffs of air) or at a point along the tube (corresponding to turbulent air at a constriction), acoustic theory tells us that the transfer function of energy from the excitation source to the output can be described in terms of the natural frequencies or resonances of the tube. Such resonances are called *formants* for speech, and they represent the frequencies that pass the most acoustic energy from the source to the output. Typically there are about three resonances of significance, for a human vocal tract, below about 3500 Hz. Figure 2.9 [5] shows a wideband spectrogram, along with the computed formant frequency estimates, for the utterance “Why do I owe you a letter,” spoken by a male speaker. There is a good correspondence between the estimated formant frequencies and the points of high spectral energy in the spectrogram. The formant frequency representation is a highly efficient, compact representation of the time-varying characteristics of speech. The major problem, however, is the difficulty of reliably estimating the formant frequencies for low-level voiced sounds, and the difficulty of defining the formants for unvoiced or silence regions. As such, this representation is more of theoretical than of practical interest.

Figures 2.10 and 2.11 show spectral and temporal representations of the phrase “Should we chase,” spoken by a male speaker, along with a detailed segmentation of the waveform into individual sounds. The ultimate goal of speech recognition is to uniquely and automatically provide such a segmentation and labeling of speech into constituent sounds or sound groups such as words, then sentences. To understand the limitations on this approach, we will next discuss, in detail, the general sounds of English and the relevant acoustic and phonetic features of the sounds.

2.4 SPEECH SOUNDS AND FEATURES

The number of linguistically distinct speech sounds (phonemes) in a language is often a matter of judgment and is not invariant to different linguists. Table 2.1 shows a condensed list of phonetic symbols of American English, their ARPABET representation [6], and an example word in which the sound occurs. Shown in this table are 48 sounds, including 18 vowels or vowel combinations (called diphthongs), 4 vowel-like consonants, 21 standard consonants, 4 syllabic sounds, and a phoneme referred to as a glottal stop (literally a symbol

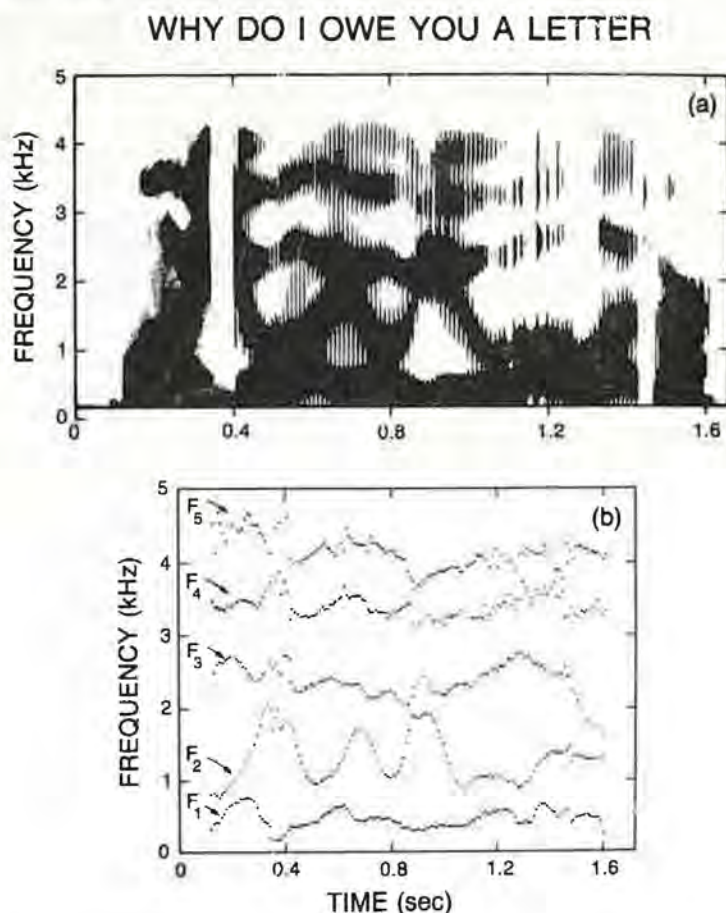


Figure 2.9 Wideband spectrogram and formant frequency representation of the utterance "Why do I owe you a letter" (after Atal and Hanauer [5]).

for a sound corresponding to a break in voicing within a sound).

Many of the sounds or phonemes shown in Table 2.1 are not considered standard; they represent specialized cases such as the so-called barred I ($/ɪ/$) in the word *roses*. As such, a more standard representation of the basic sounds and sound classes of American English is shown in Figure 2.12. Here we see the conventional set of 11 vowels, classified as front, mid, or back, corresponding to the position of the tongue hump in producing the vowel; 4 vowel combinations or diphthongs; the 4 semivowels broken down into 2 liquids and 2 glides; the nasal consonants, the voiced and unvoiced stop consonants; the voiced and unvoiced fricatives; whisper; and the affricates. There are a total of 39 of the 48 sounds of Table 2.1 represented in Figure 2.12.

2.4.1 The Vowels

The vowel sounds are perhaps the most interesting class of sounds in English. Their importance to the classification and representation of written text is very low; however, most practical speech-recognition systems rely heavily on vowel recognition to achieve

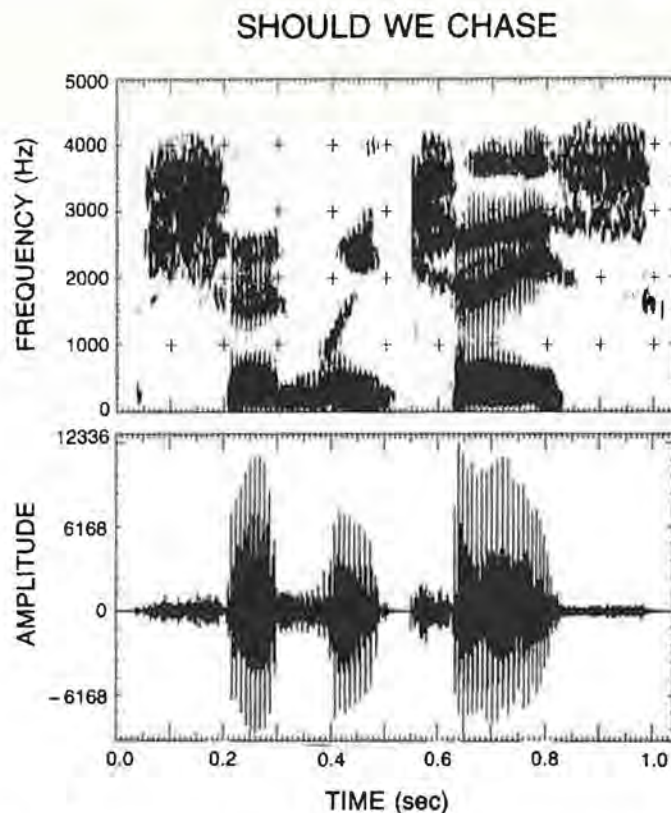


Figure 2.10 Wideband spectrogram and intensity contour of the phrase “Should we chase.”

high performance. To partially illustrate this point, consider the following sections of text:

Section I

Th_y n_t_d s_gn_f_c_nt _mpr_v_m_nts i_ th_ c_mp_ny's _m_g_ s_p_rv_s__n,
th__r w_rk_ng c_nd_t__ns, b_n_f_ts _nd _pp_rt_n_t_s f_r gr_wth.

Section II

A__i_u_e__o_a__a__a_e_e__e_ia__ __e_a_e, _i__ __e__o_e_o__
o__u_a_io_a_e__o_ee__i____e_ea_i__.

In Section I we have omitted the conventional vowel letters (a,e,i,o,u); however, with a little effort the average reader can “fill in” the missing vowels and decode the section so that it reads

They noted significant improvements in the company's image, supervision,
their working conditions, benefits and opportunities for growth.

In Section II we have omitted the conventional consonant letters; the resulting text is essentially not decodable. The actual text is

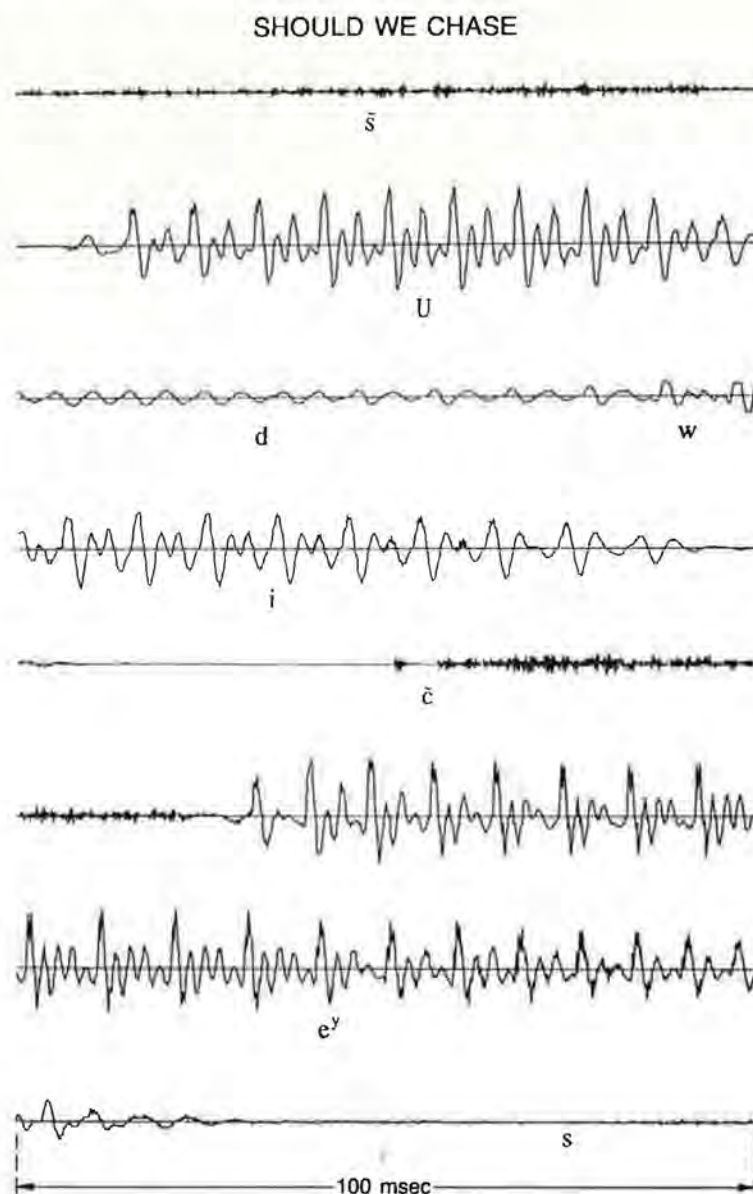


Figure 2.11 The speech waveform and a segmentation and labeling of the constituent sounds of the phrase "Should we chase."

Attitudes toward pay stayed essentially the same, with the scores of occupational employees slightly decreasing.

In speaking, vowels are produced by exciting an essentially fixed vocal tract shape with quasi-periodic pulses of air caused by the vibration of the vocal cords. The way in which the cross-sectional area varies along the vocal tract determines the resonance frequencies of the tract (the formants) and thereby the sound that is produced. The vowel sound produced is determined primarily by the position of the tongue, but the positions of the jaw, lips, and to a small extent, the velum, also influence the resulting sound.

TABLE 2.1. A condensed list of phonetic symbols for American English.

Phoneme	ARPABET	Example	Phoneme	ARPABET	Example
/i/	IY	beat	/ŋ/	NX	sing
/I/	IH	bit	/p/	P	pet
/e/ (e ^y)	EY	bait	/t/	T	ten
/ɛ/	EH	bet	/k/	K	kit
/æ/	AE	bat	/b/	B	bet
/a/	AA	Bob	/d/	D	debt
/ʌ/	AH	but	/g/	H	get
/ɔ/	AO	bought	/h/	HH	hat
/o/ (o ^w)	OW	boat	/f/	F	fat
/U/	UH	book	/θ/	TH	thing
/u/	UW	boot	/s/	S	sat
/ə/	AX	about	/ʃ/ (sh)	SH	shut
/ɪ/	IX	roses	/v/	V	yat
/ɜ/	ER	bird	/ð/	DH	that
/ɔ/	AXR	butter	/z/	Z	zoo
/a ^w /	AW	down	/ʒ/ (zh)	ZH	azure
/a ^y /	AY	buy	/tʃ/ (tsh)	CH	church
/ɔ ^y /	OY	boy	/dʒ/ (dzh, j)	JH	judge
/y/	Y	you	/w/	WH	which
/w/	W	wit	/l/	EL	battle
/r/	R	rent	/ŋ/	EM	bottom
/l/	L	let	/n/	EN	button
/m/	M	met	/T/	DX	batter
/n/	N	net	/ʔ/	Q	(glottal stop)

The vowels are generally long in duration (as compared to consonant sounds) and are spectrally well defined. As such they are usually easily and reliably recognized and therefore contribute significantly to our ability to recognize speech, both by human beings and by machine.

There are several ways to characterize and classify vowels, including the typical articulatory configuration required to produce the sounds, typical waveform plots, and typical spectrogram plots. Figures 2.13–2.15 show typical articulatory configurations of the vowels (2.13), examples of vowel waveforms (2.14), and examples of vowel spectrograms (2.15). A convenient and simplified way of classifying vowel articulatory configurations is in terms of the tongue hump position (i.e., front, mid, back), and tongue hump height (high, mid, low), where the tongue hump is the mass of the tongue at its narrowest constriction within the vocal tract. According to this classification the vowels /i/, /I/, /æ/, and /ɛ/ are front vowels, (with different tongue heights) /a/, /ʌ/, and /ɔ/ are mid vowels, and /U/, /u/, and /o/ are back vowels (see also Figure 2.12).

As shown in the acoustic waveform plots of the vowels, in Figure 2.14, the front vowels show a pronounced, high-frequency resonance, the mid vowels show a balance of energy over a broad frequency range, and the back vowels show a predominance of low-frequency spectral information. This behavior is evidenced in the vowel spectrogram

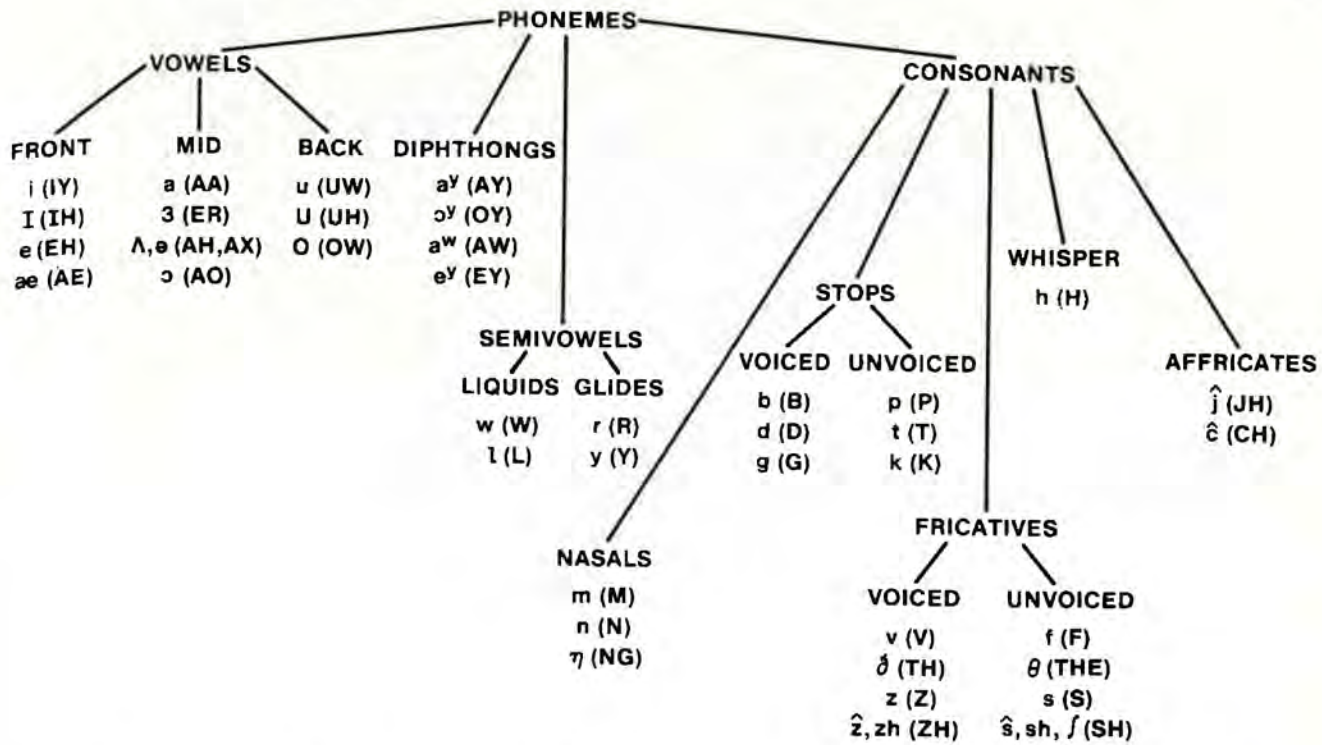


Figure 2.12 Chart of the classification of the standard phonemes of American English into broad sound classes.

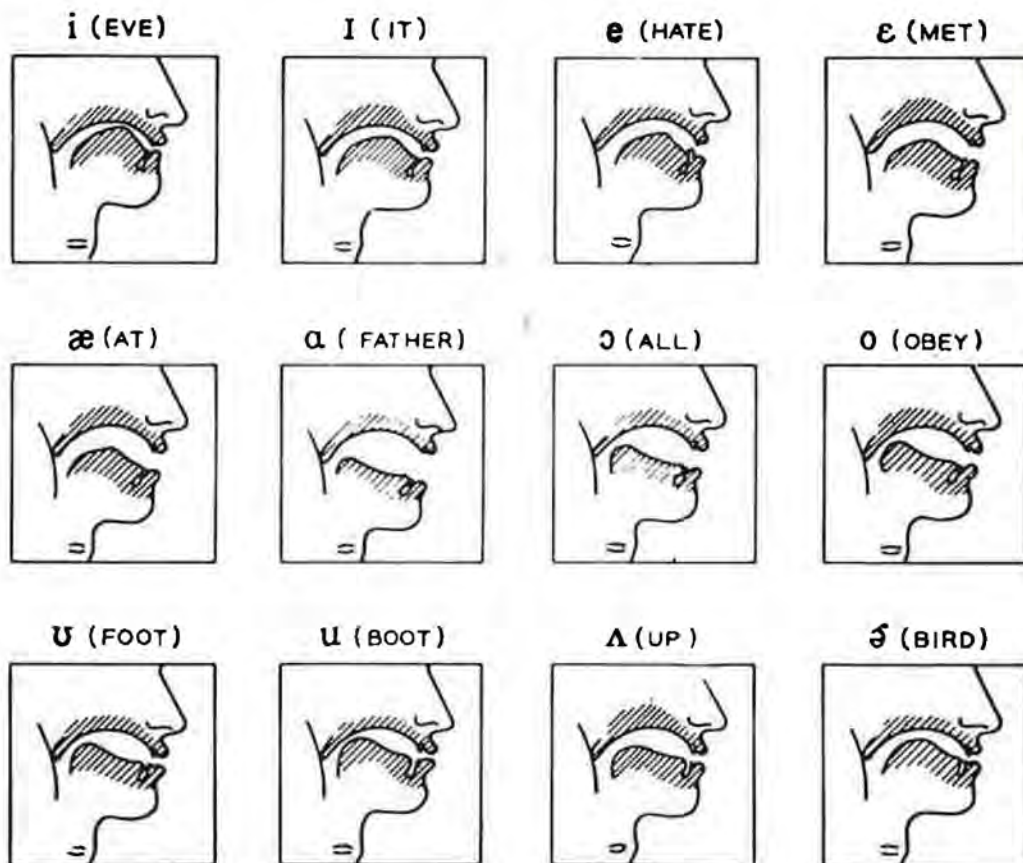


Figure 2.13 Articulatory configurations for typical vowel sounds (after Flanagan [3]).

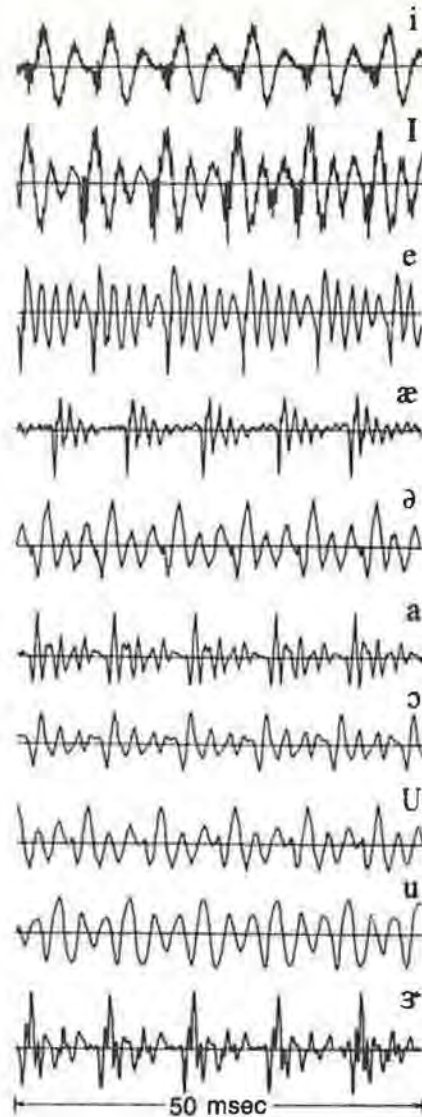


Figure 2.14 Acoustic waveform plots of typical vowel sounds.

plots of Figure 2.15, in which the front vowels show a relatively high second and third formant frequency (resonance), whereas the mid vowels show well-separated and balanced locations of the formants, and the back vowels (especially /u/) show almost no energy beyond the low-frequency region with low first and second formant frequencies.

The concept of a “typical” vowel sound is, of course, unreasonable in light of the variability of vowel pronunciation among men, women and children with different regional accents and other variable characteristics. To illustrate this point, Figure 2.16 shows a classic plot, made by Gordon Peterson and Harold Barney, of measured values of the first and second formant for 10 vowels spoken by a wide range of male and female talkers who attended the 1939 World’s Fair in New York City [7]. A wide range of variability can be seen in the measured formant frequencies for a given vowel sound, and also there is

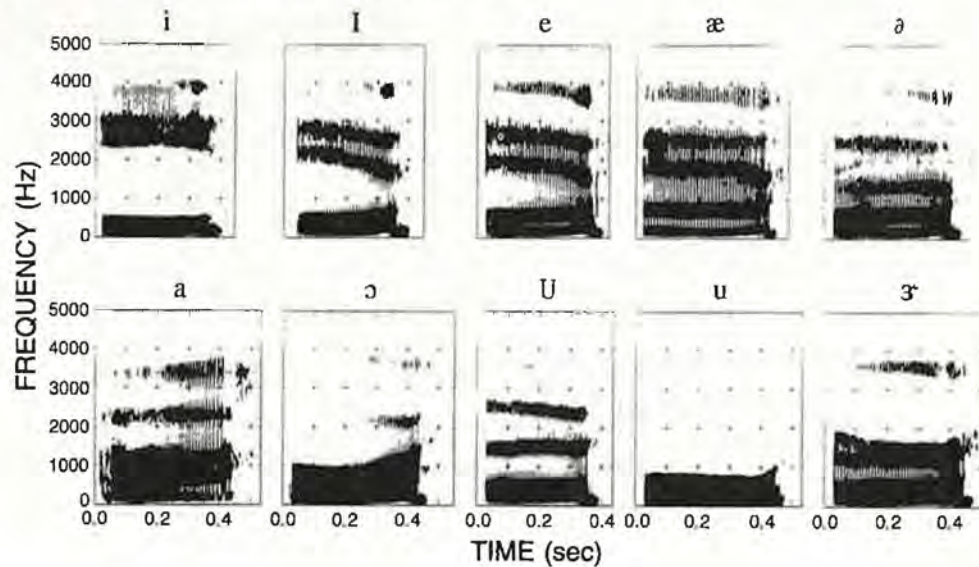


Figure 2.15 Spectrograms of the vowel sounds.

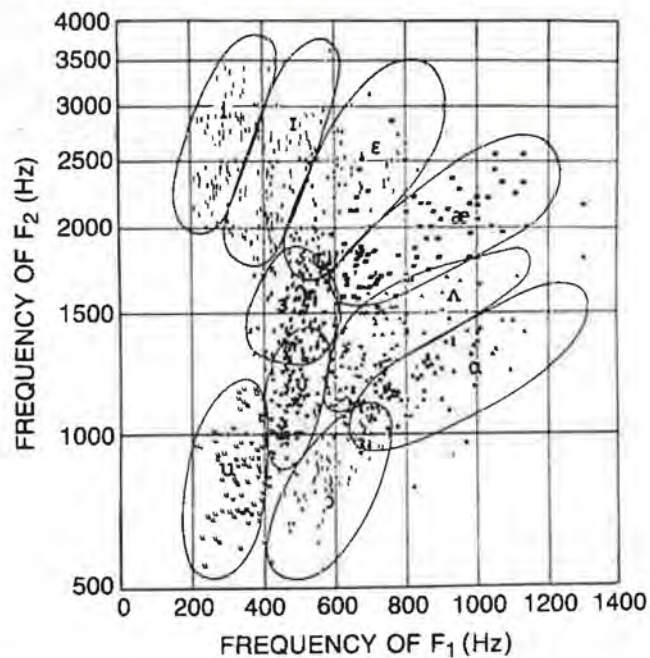


Figure 2.16 Measured frequencies of first and second formants for a wide range of talkers for several vowels (after Peterson & Barney [7]).

overlap between the formant frequencies for *different* vowel sounds by different talkers. The ellipses drawn in this figure represent gross characterizations of the regions in which most of the tokens of the different vowels lie. The message of Figure 2.16, for speech recognition by machine, is fairly clear; that is, it is not just a simple matter of measuring formant frequencies or spectral peaks accurately to accurately classify vowel sounds; one

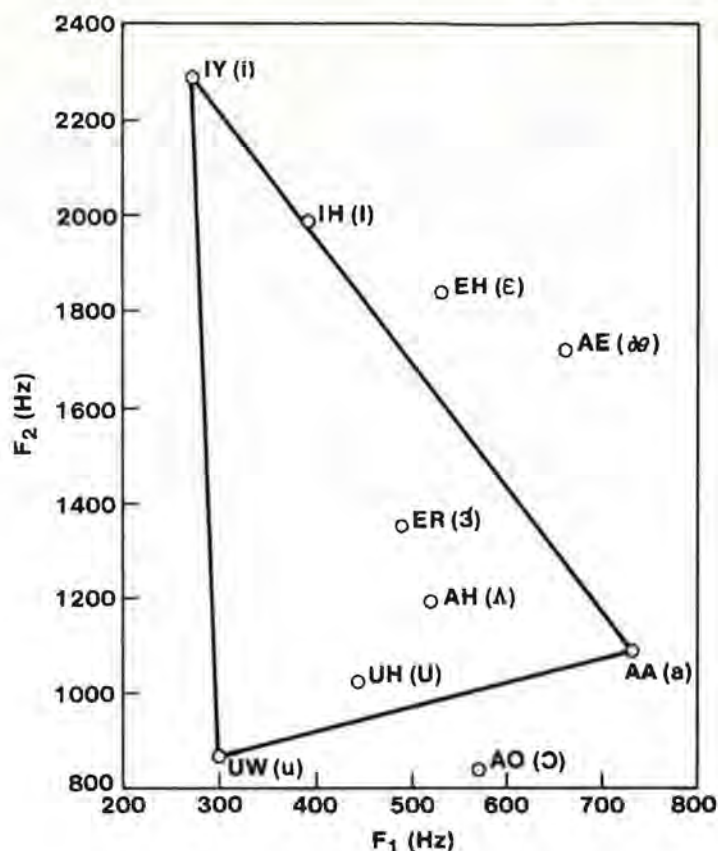


Figure 2.17 The vowel triangle with centroid positions of the common vowels.

must do some type of talker (accent) normalization to account for the variability in formants and overlap between vowels.

A common way of exploiting the information embodied in Figures 2.15 and 2.16 is to represent each vowel by a centroid in the formant space with the realization that the centroid, at best, represents average behavior and does not represent variability across talkers. Such a representation leads to the classic vowel triangle shown in Figure 2.17 and represented in terms of formant positions by the data given in Table 2.2. The vowel triangle represents the extremes of formant locations in the F_1 - F_2 plane, as represented by /i/ (low F_1 , high F_2), /u/ (low F_1 , low F_2), and /a/ (high F_1 , low F_2), with other vowels appropriately placed with respect to the triangle vertices. The utility of the formant frequencies of Table 2.2 has been demonstrated in text-to-speech synthesis in which high-quality vowel sounds have been synthesized using these positions for the resonances [8].

2.4.2 Diphthongs

Although there is some ambiguity and disagreement as to what is and what is not a diphthong, a reasonable definition is that a diphthong is a gliding monosyllabic speech

TABLE 2.2. Formant frequencies for typical vowels.

ARPABET Symbol for Vowel	IPA Symbol	Typical Word	F ₁	F ₂	F ₃
IY	/i/	beet	270	2290	3010
IH	/ɪ/	bit	390	1990	2550
EH	/ɛ/	bet	530	1840	2480
AE	/æ/	bat	660	1720	2410
AH	/ʌ/	but	520	1190	2390
AA	/ɑ/	hot	730	1090	2440
AO	/ɔ/	bought	570	840	2410
UH	/ʊ/	foot	440	1020	2240
UW	/u/	boot	300	870	2240
ER	/ɜ/	bird	490	1350	1690

sound that starts at or near the articulatory position for one vowel and moves to or toward the position for another. According to this definition, there are six diphthongs in American English, namely /aʏ/ (as in buy), /aʷ/ (as in down), /eʏ/ (as in bait), and /ɔʏ/ (as in boy), /o/ (as in boat), and /ju/ (as in you).

The diphthongs are produced by varying the vocal tract smoothly between vowel configurations appropriate to the diphthong. Figure 2.18 shows spectrogram plots of four of the diphthongs spoken by a male talker. The gliding motions of the formants are especially prominent for the sounds /aʏ/, /aʷ/ and /ɔʏ/ and are somewhat weaker for /eʏ/ because of the closeness (in vowel space) of the two vowel sounds comprising this diphthong.

An alternative way of displaying the time-varying spectral characteristics of diphthongs is via a plot of the values of the second formant versus the first formant (implicitly as a function of time) as shown in Figure 2.19 [9]. The arrows in this figure indicate the direction of motion of the formants (in the (F₁ – F₂) plane) as time increases. The dashed circles in this figure indicate average positions of the vowels. Based on these data, and other measurements, the diphthongs can be characterized by a time-varying vocal tract area function that varies between two vowel configurations.

2.4.3 Semivowels

The group of sounds consisting of /w/, /l/, /r/, and /y/ is quite difficult to characterize. These sounds are called semivowels because of their vowel-like nature. They are generally characterized by a gliding transition in vocal tract area function between adjacent phonemes. Thus the acoustic characteristics of these sounds are strongly influenced by the context in which they occur. For our purposes, they are best described as transitional, vowel-like sounds, and hence are similar in nature to the vowels and diphthongs.

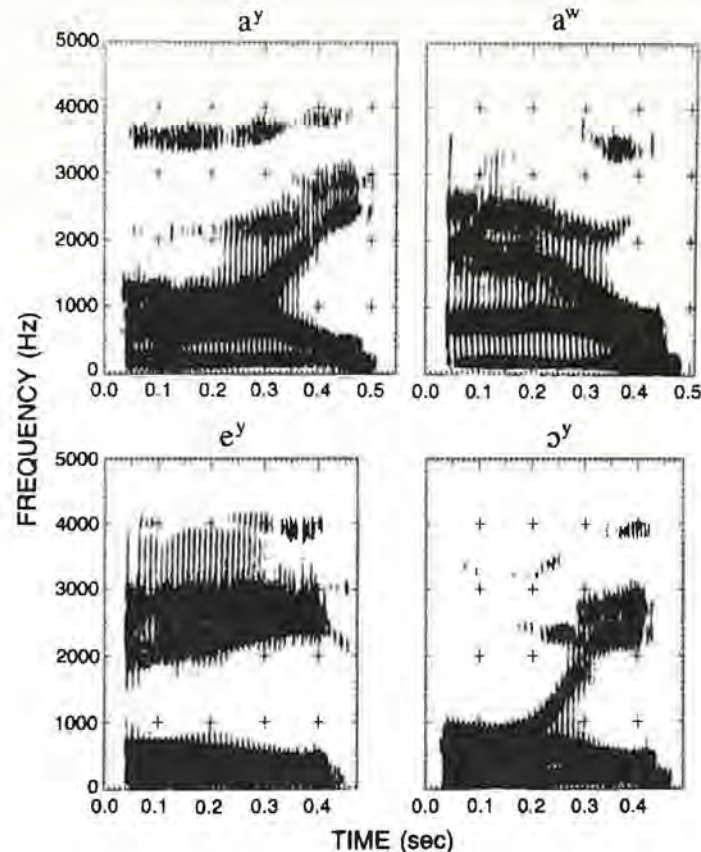


Figure 2.18 Spectrogram plots of four diphthongs.

2.4.4 Nasal Consonants

The nasal consonants /m/, /n/, and /ŋ/ are produced with glottal excitation and the vocal tract totally constricted at some point along the oral passageway. The velum is lowered so that air flows through the nasal tract, with sound being radiated at the nostrils. The oral cavity, although constricted toward the front, is still acoustically coupled to the pharynx. Thus, the mouth serves as a resonant cavity that traps acoustic energy at certain natural frequencies. As far as the radiated sound is concerned, these resonant frequencies of the oral cavity appear as antiresonances, or zeros of the transfer function of sound transmission. Furthermore, nasal consonants and nasalized vowels (i.e., some vowels preceding or following nasal consonants) are characterized by resonances that are spectrally broader, or more highly damped, than those for vowels.

The three nasal consonants are distinguished by the place along the oral tract at which a total constriction is made. For /m/ the constriction is at the lips; for /n/ the constriction is just behind the teeth; and for /ŋ/ the constriction is just forward of the velum itself. Figure 2.20 shows typical speech waveforms and Figure 2.21 spectrograms for two nasal consonants in the context vowel-nasal-vowel. The waveforms of /m/ and /n/ look very similar. The spectrograms show a concentration of low-frequency energy with a midrange of frequencies that contain no prominent peaks. This is because of the particular

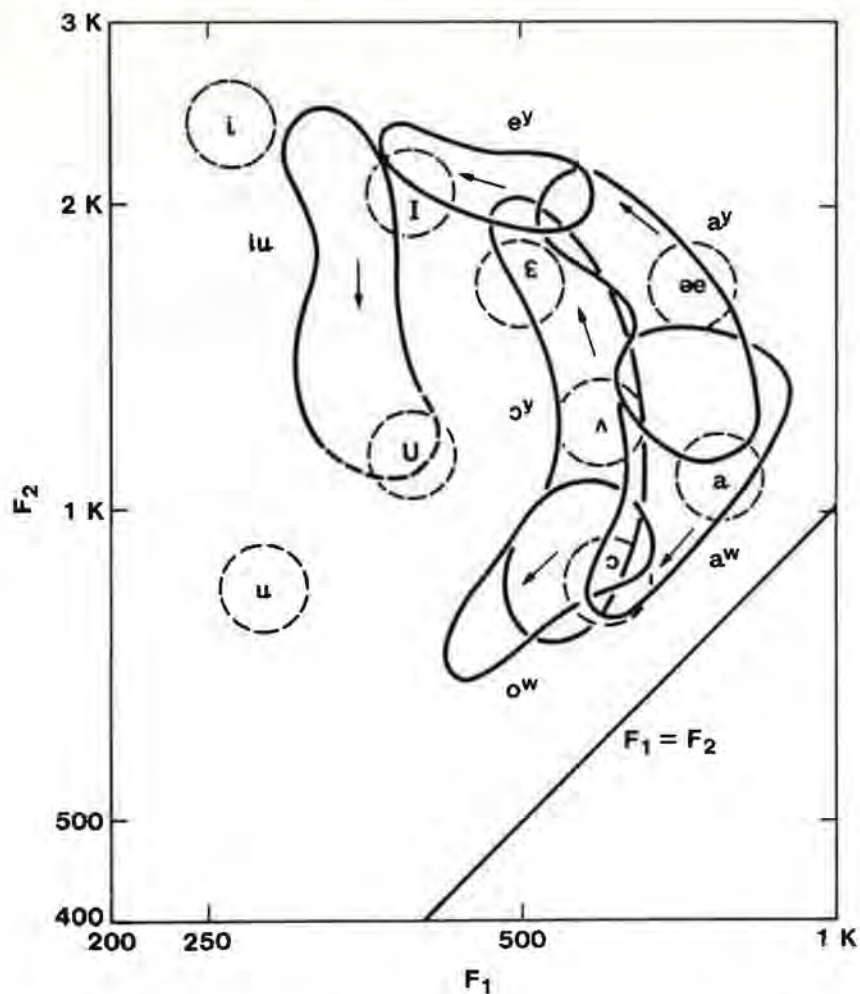


Figure 2.19 Time variation of the first two formants for the diphthongs (after Holbrook and Fairbanks [9]).

combination of resonances and antiresonances that result from the coupling of the nasal and oral tracts.

2.4.5 Unvoiced Fricatives

The unvoiced fricatives /f/, /θ/, /s/, and /sh/ are produced by exciting the vocal tract by a steady air flow, which becomes turbulent in the region of a constriction in the vocal tract. The location of the constriction serves to determine which fricative sound is produced. For the fricative /f/ the constriction is near the lips; for /θ/ it is near the teeth; for /s/ it is near the middle of the oral tract; and for /sh/ it is near the back of the oral tract. Thus the system for producing unvoiced fricatives consists of a source of noise at a constriction, which separates the vocal tract into two cavities. Sound is radiated from the lips—that is, from the front cavity. The back cavity serves, as in the case of nasals, to trap energy and thereby introduce antiresonances into the vocal output. Figure 2.22 shows the waveforms and Figure 2.23 the spectrograms of the fricatives /f/, /s/ and /sh/. The nonperiodic nature

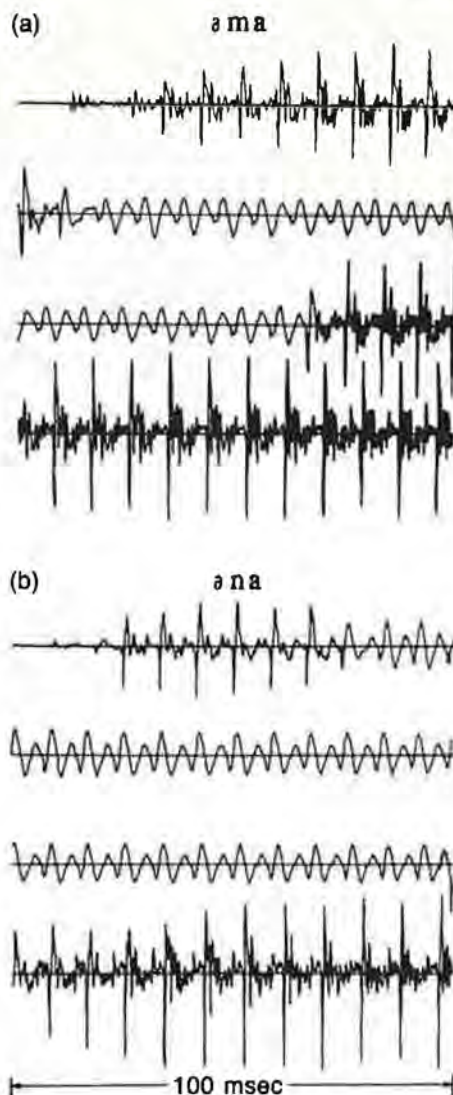


Figure 2.20 Waveforms for the sequences /a-m-a/ and /a-n-a/.

of fricative excitation is obvious in the waveform plots. The spectral differences among the fricatives are readily seen by comparing the three spectrograms.

2.4.6 Voiced Fricatives

The voiced fricatives /v/, /th/, /z/ and /zh/ are the counterparts of the unvoiced fricatives /f/, /θ/, /s/, and /sh/, respectively, in that the place of constriction for each of the corresponding phonemes is essentially identical. However, the voiced fricatives differ markedly from their unvoiced counterparts in that two excitation sources are involved in their production. For voiced fricatives the vocal cords are vibrating, and thus one excitation source is at the glottis. However, since the vocal tract is constricted at some point forward of the glottis, the air flow becomes turbulent in the neighborhood of the constriction. Thus the

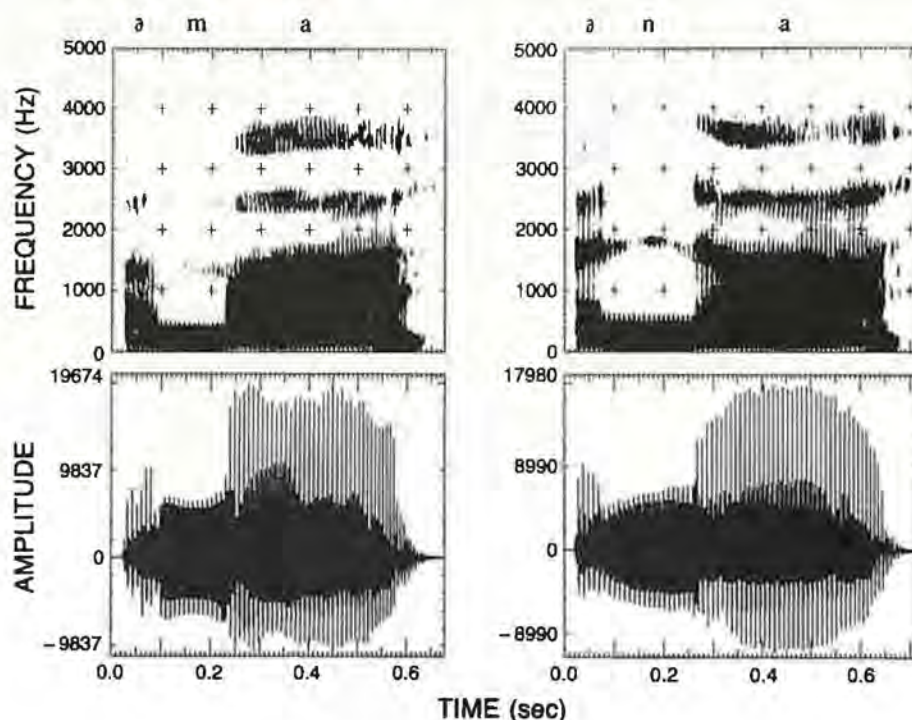


Figure 2.21 Spectrograms of the sequences /a-m-a/ and a-n-a/.

spectra of voiced fricatives can be expected to display two distinct components. These excitation features are readily observable in Figure 2.24, which shows typical waveforms, and in Figure 2.25, which shows spectra for two voiced fricatives. The similarity of the unvoiced fricative /f/ to the voiced fricative /v/ is easily shown in a comparison between corresponding spectrograms in Figures 2.23 and 2.25. Likewise, it is instructive to compare the spectrograms of /sh/ and /zh/.

2.4.7 Voiced and Unvoiced Stops

The voiced stop consonants /b/, /d/, and /g/, are transient, noncontinuant sounds produced by building up pressure behind a total constriction somewhere in the oral tract and then suddenly releasing the pressure. For /b/ the constriction is at the lips; for /d/ the constriction is at the back of the teeth; and for /g/ it is near the velum. During the period when there is total constriction in the tract, no sound is radiated from the lips. However, there is often a small amount of low-frequency energy radiated through the walls of the throat (sometimes called a voice bar). This occurs when the vocal cords are able to vibrate even though the vocal tract is closed at some point.

Since the stop sounds are dynamical in nature, their properties are highly influenced by the vowel that follows the stop consonant. As such, the waveforms for stop consonants give little information about the particular stop consonant. Figure 2.26 shows the waveform of the syllable /a-b-a/. The waveform of /b/ shows few distinguishing features except for the voiced excitation and lack of high-frequency energy.

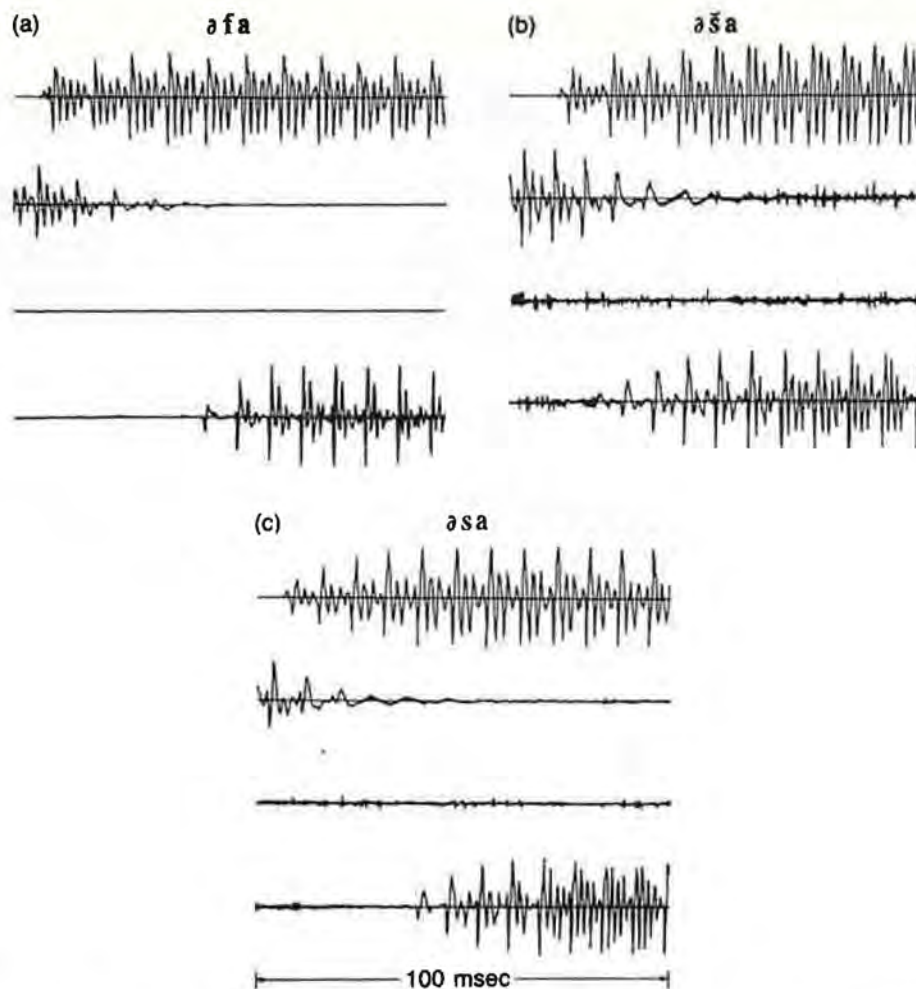


Figure 2.22 Waveforms for the sounds /f/, /s/ and /sh/ in the context /ə-x-a/ where /x/ is the unvoiced fricative.

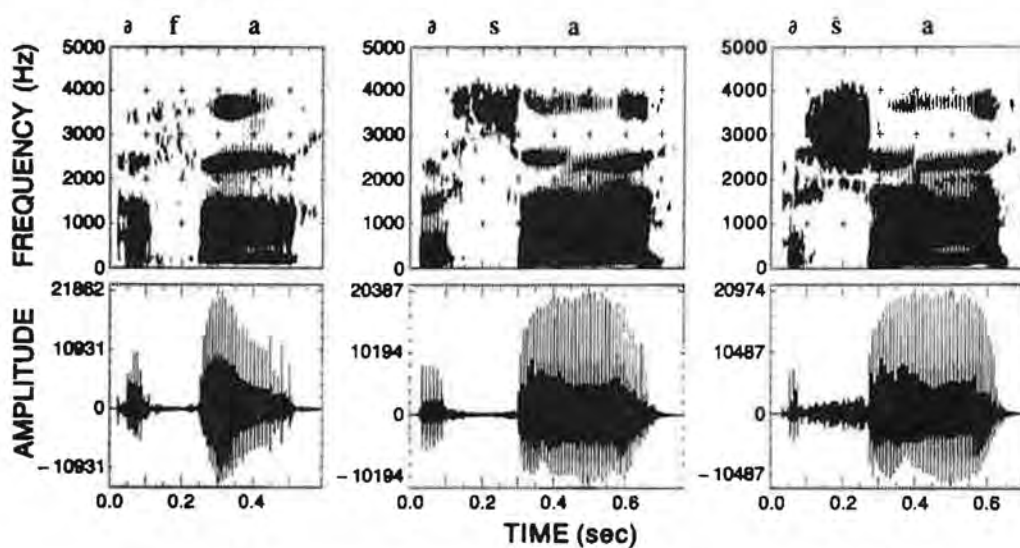


Figure 2.23 Spectrogram comparisons of the sounds /ə-f-a/, /ə-s-a/ and /ə-sh-a/.

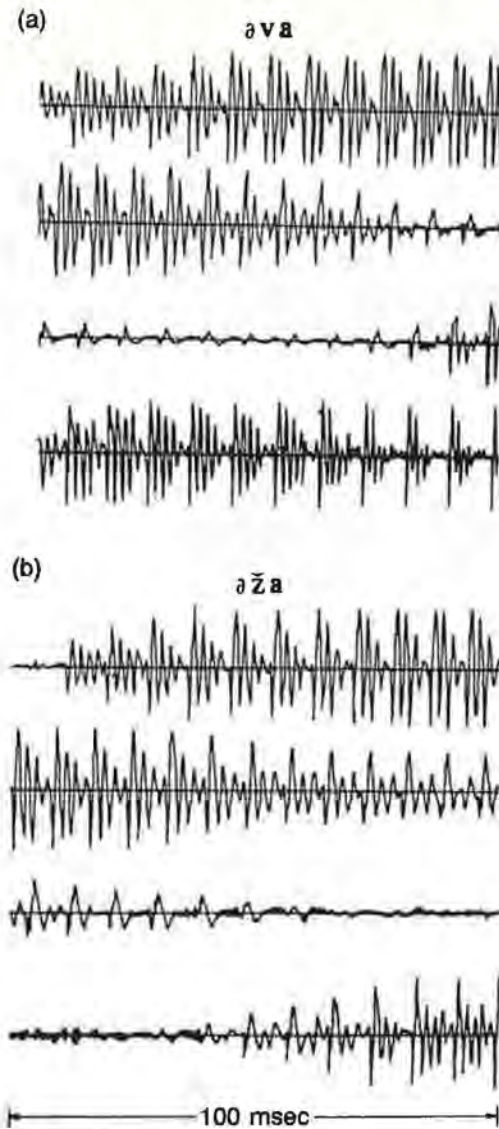


Figure 2.24 Waveforms for the sequences /əv-a/ and /ə-zh-a/.

The unvoiced stop consonants /p/, /t/, and /k/ are similar to their voiced counterparts /b/, /d/, and /g/, with one major exception. During the period of total closure of the tract, as the pressure builds up, the vocal cords do not vibrate. Then, following the period of closure, as the air pressure is released, there is a brief interval of friction (due to sudden turbulence of the escaping air) followed by a period of aspiration (steady air flow from the glottis exciting the resonances of the vocal tract) before voiced excitation begins.

Figure 2.27 shows waveforms and Figure 2.28 shows spectrograms of the voiced stop /b/ and the voiceless stop consonants /p/ and /t/. The “stop gap,” or time interval, during which the pressure is built up is clearly in evidence. Also, it can be readily seen that the duration and frequency content of the friction noise and aspiration vary greatly with the stop consonant.

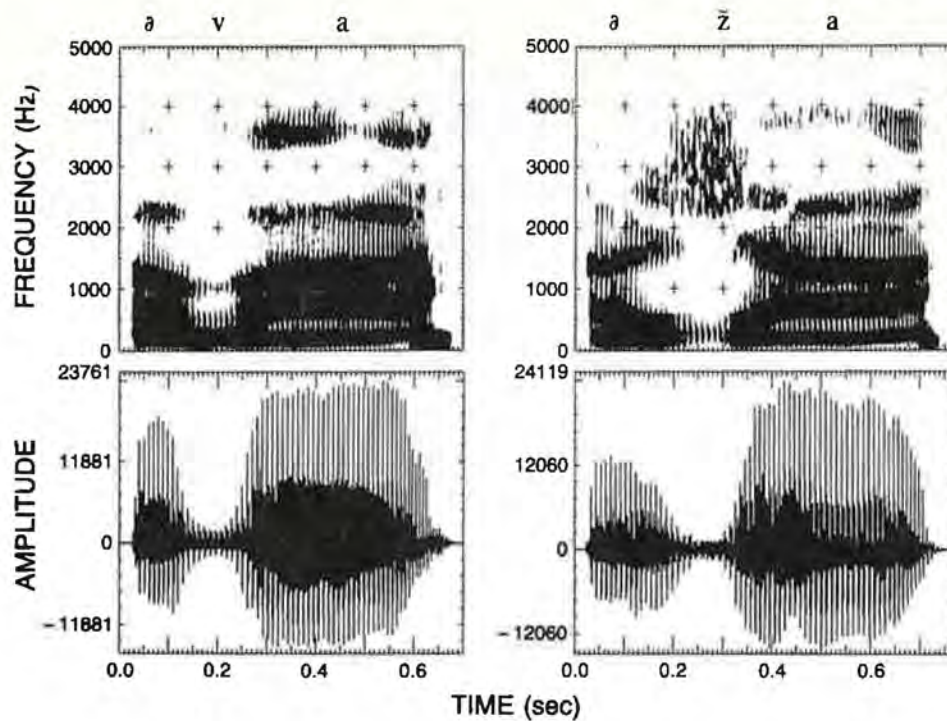


Figure 2.25 Spectrograms for the sequences /ə-v-a/ and /ə-zh-a/.

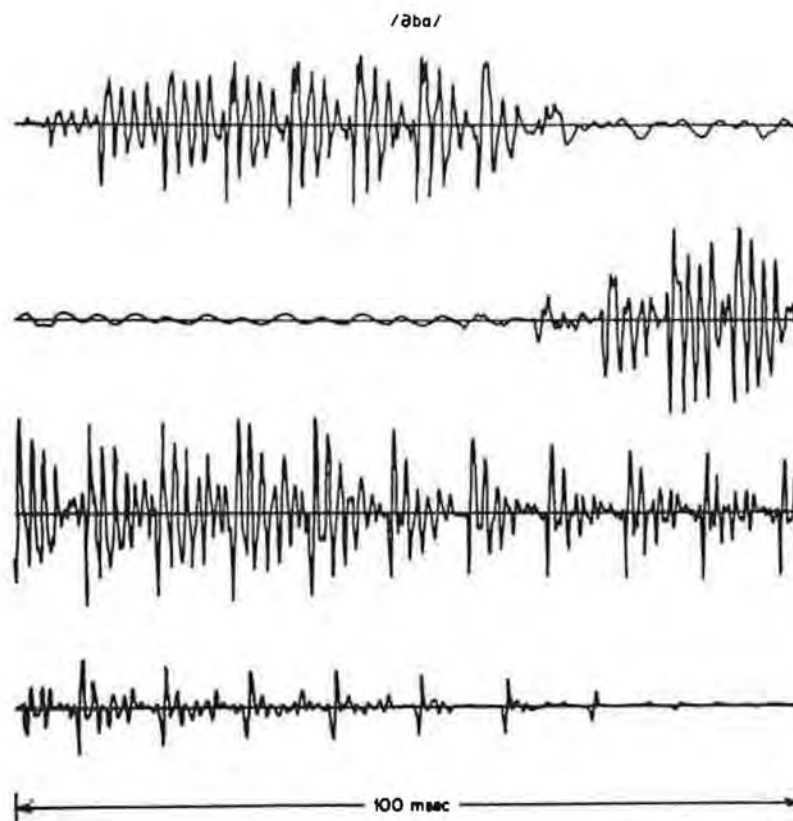


Figure 2.26 Waveform for the sequence /ə-b-a/.

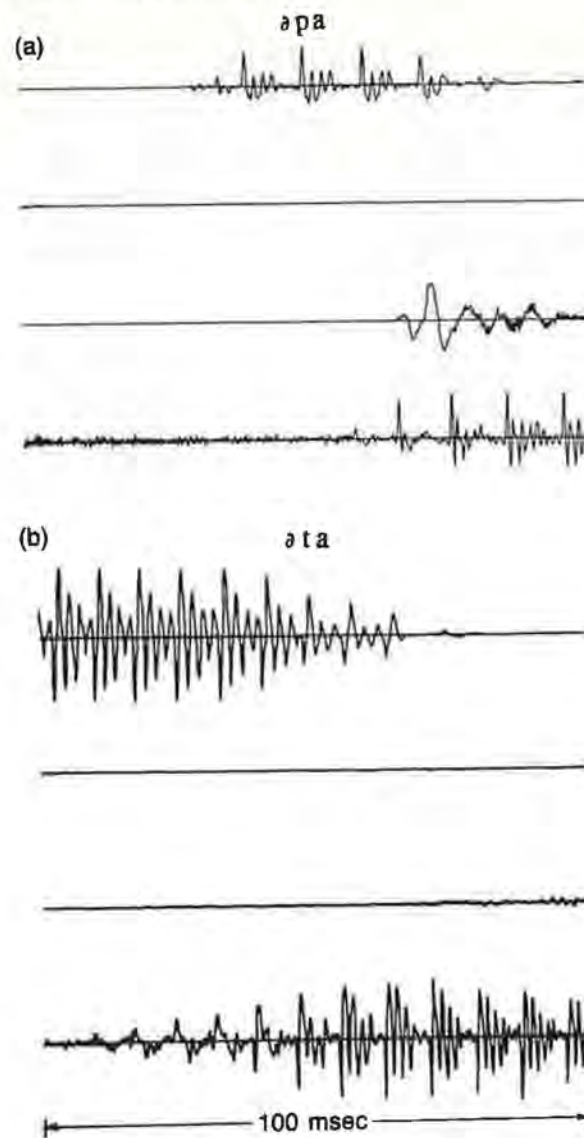


Figure 2.27 Waveforms for the sequences /ə-p-a/ and /ə-t-a/.

2.4.8 Review Exercises

As a self-check on the reader's understanding of the material on speech sounds and their acoustic manifestations, we now digress and present some simple exercises along with the solutions. For maximum effectiveness, the reader is encouraged to think through each exercise before looking at the solution.

Exercise 2.1

1. Write out the phonetic transcription for the following words:
he, eats, several, light, tacos
2. What effect occurs when these five words are spoken in sequence as a sentence? What does this imply about automatic speech recognition?

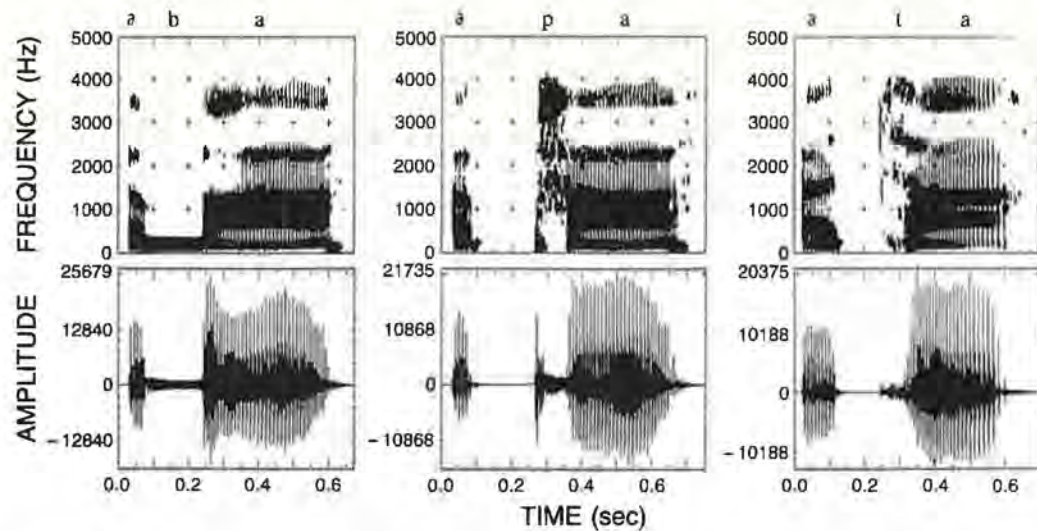


Figure 2.28 Spectrogram comparisons of the sequences of voiced (/ə-b-a/) and voiceless (/ə-p-a/ and /ə-t-a/) stop consonants.

Solution 2.1

1. The phonetic transcriptions of the words are

Word	Phoneme Sequence	ARPABET
he	/hi/	HH-IY
eats	/its/	IY-TS
several	/sɛv rəl/	S-EH-V-R-AH-L
light	/laɪt/	L-AY-T
tacos	/takoz/	T-AA-K-OW-Z

2. When the words are spoken together, the last sound of each word merges with the first sound of the succeeding word (since they are the same sound), resulting in strong coarticulation of boundary sounds. The ARPABET transcription for the sentence is:

HH-IY-T-S-EH-V-R-AH-L-AY-T-AA-K-OW-Z

All information about word boundaries is totally lost; furthermore, the durations of the common sounds at the boundaries of words are much shorter than what would be predicted from the individual words.

Exercise 2.2

Some of the difficulties in large vocabulary speech recognition are related to the irregularities in the way basic speech sounds are combined to produce words. Exercise 2.2 highlights a couple of these difficulties.

1. In word initial position of American English, which phoneme or phonemes can never occur? Which hardly ever occur?
2. There are many word initial consonant clusters of length two, such as *speak*, *drank*, *plead*, and *press*. How many word initial consonant clusters of length three are there in American English? What general rule can you give about the sounds in each of the three positions?

3. A nasal consonant can be combined with a stop consonant (e.g., *camp*, *tend*) in a limited number of ways. What general rule do such combinations obey? There are several notable exceptions to this general rule. Can you give a couple of exceptions? What kind of speaking irregularity often results from these exceptions?

Solution 2.2

1. The only phoneme that never occurs in initial word position in English is the /ng/ sound (e.g., *sing*). The only other sound that almost never occurs naturally in English, in initial word position, is /zh/ except some foreign words imported into English, such as *gendarme*, which does have an initial /zh/.
2. The word initial consonant clusters of length three in English include

/spl/	—	split
/spr/	—	spring
/skw/	—	squirt
/skr/	—	script
/str/	—	string

The general rule for such clusters is

/sound s/unvoiced stop/semivowel/

3. The general rule for a nasal-stop combination is that the nasal and stop have the same place of articulation, e.g., front/lips (/mp/), mid/dental (/nt/), back/velar (/ng k/). Exceptions occur in words like *summed* (/md/) or *hanged* (/ng d/) or *dreamt* (/mt/). There is often a tendency to insert an extra stop in such situations (e.g., *dreamt* → /drempt/).

Exercise 2.3

An important speech task is accurate digit recognition. This exercise seeks to exploit knowledge of acoustic phonetics to recognize first isolated digits, and next some simple connected digit strings. We first need a sound lexicon (a dictionary) for the digits. The sound lexicon describes the pronunciations of digits in terms of the basic sounds of English. Such a sound lexicon is given in Table 2.3. A single male adult talker (LRR) spoke each of the 11 digits in random sequence and in isolation, and spectrograms of these spoken utterances are shown in Figure 2.29. Figure 2.30 shows spectrograms of two connected digit sequences spoken by the same talker.

1. Identify each of the 11 digits based on the acoustic properties of the sounds within the digit (as expressed in the sound lexicon). Remember that each digit was spoken exactly once.
2. Try to identify the spoken digits in each of the connected digit strings.

Solution 2.3

1. The digits of the top row are 3 and 7:
 - a. The digit 3 is cued by the distinctive brief initial fricative (/θ/), followed by the semivowel /r/ where the second and third formants both get very low in frequency, followed by the /i/ where F_2 and F_3 both become very high in frequency.
 - b. The digit 7 is cued by the strong /s/ frication at the beginning, the distinctive /ε/, followed by the voiced fricative /v/, a short vowel /ə/ and ending in the strong

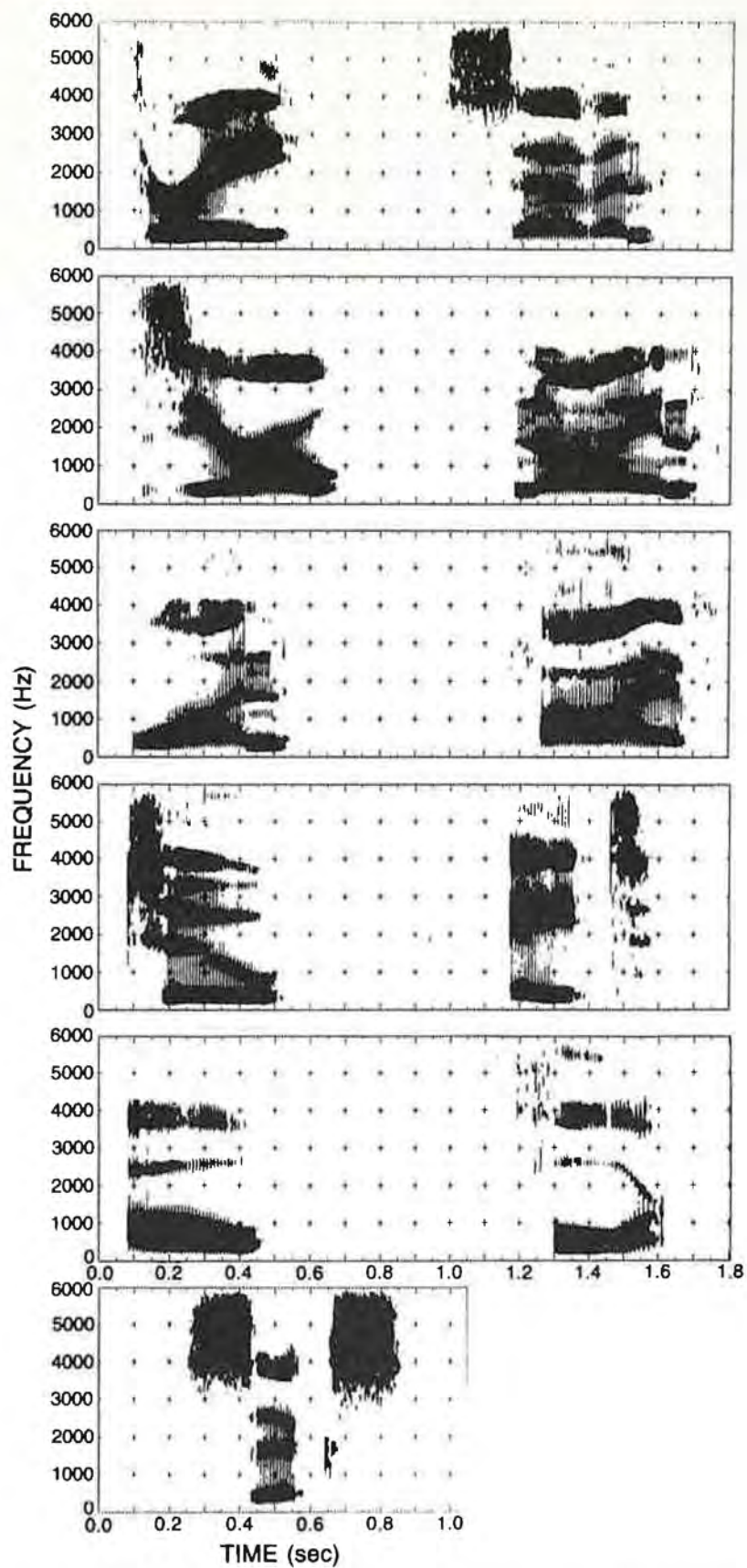
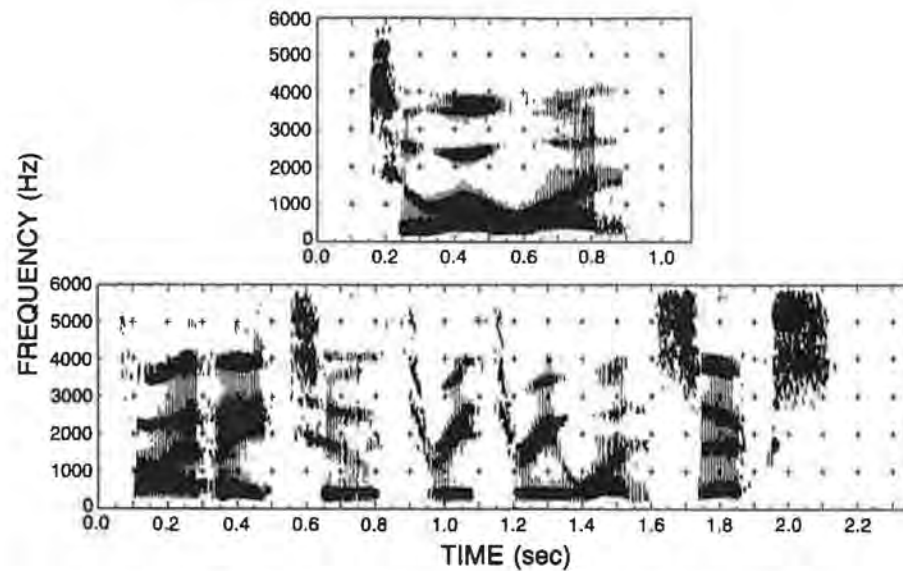


Figure 2.29 Spectrograms of the 11 isolated digits, 0 through 9 plus oh, in random sequence.

TABLE 2.3. Sound Lexicon of Digits

Word	Sounds	ARPABET
Zero	/z ɪ r o/	Z-IH-R-OW
One	/w ʌ n/	W-AH-N
Two	/t u/	T-UW
Three	/θ r i/	TH-R-IY
Four	/f o r/	F-OW-R
Five	/f aʲ v/	F-AY-V
Six	/s ɪ k s/	S-IH-K-S
Seven	/s ɛ v ə n/	S-EH-V-AX-N
Eight	/eʲ t/	EY-T
Nine	/n aʲ n/	N-AY-N
Oh	/o/	OW

**Figure 2.30** Spectrograms of two connected digit sequences.

nasal /n/.

The digits in the second row are 0 and 9:

- a. The initial /z/ is cued by the strong frication with the presence of voicing at low frequencies; the following /l/ is seen by the high F_2 and F_3 , the /r/ is signaled by the low F_2 and F_3 , and the diphthong /o/ is signaled by the gliding motion of F_2 and F_3 toward an /u/-like sound.
- b. The digit 9 is cued by the distinct initial and final nasals /n/ and by the /aʲ/ glide between the nasals.

The digits in the third row are 1 and 5:

- a. The digit 1 is cued by the strong initial semivowel /w/ with very low F_2 and by the strong final nasal /n/.
- b. The digit 5 is cued by the weak initial frication of /f/, followed by the strong diphthong /aʲ/ and ending in the very weak fricative /v/.

The digits in the fourth row are 2 and 8:

- a. The digit 2 is cued by the strong /t/ burst and release followed by the glide to the /u/ sound.
- b. The digit 8 is cued by the initial weak diphthong /e^y/ followed by a clear stop gap of the /t/ and then the /t/ release.

The digits in the fifth row are “oh” and 4:

- a. The digit “oh” is virtually a steady sound with a slight gliding tendency toward /u/ at the end.
- b. The digit 4 is cued by the weak initial fricative /f/, followed by the strong /o/ vowel and ending with a classic /r/ where F_2 and F_3 merge together.

The digit in the last row is 6:

- a. The digit 6 is cued by the strong /s/ frication at the beginning and end, and by the steady vowel /l/ followed by the stop gap and release of the /k/.
2. By examining the isolated digit sequences, one can eventually (with a lot of work and some good luck) conclude that the two sequences are

Row 1: 2-oh-1 (telephone area code)

Row 2: 5-8-2-3-3-1-6 (7-digit telephone number)

We will defer any explanation of how any reasonable person, or machine, could perform this task until later in this book when we discuss connected word-recognition techniques. The purpose of this exercise is to convince the reader how difficult a relatively simple recognition task can be.

2.5 APPROACHES TO AUTOMATIC SPEECH RECOGNITION BY MACHINE

The material presented in the previous sections leads to a straightforward way of performing speech recognition by machine whereby the machine attempts to decode the speech signal in a sequential manner based on the observed acoustic features of the signal and the known relations between acoustic features and phonetic symbols. This method, appropriately called the acoustic-phonetic approach, is indeed viable and has been studied in great depth for more than 40 years. However, for a variety of reasons, the acoustic-phonetic approach has not achieved the same success in practical systems as have alternative methods. Hence, in this section, we provide an overview of several proposed approaches to automatic speech recognition by machine with the goal of providing some understanding as to the essentials of each proposed method, and the basic strengths and weaknesses of each approach.

Broadly speaking, there are three approaches to speech recognition, namely:

1. the acoustic-phonetic approach
2. the pattern recognition approach
3. the artificial intelligence approach

The acoustic-phonetic approach is based on the theory of acoustic phonetics that postulates that there exist finite, distinctive phonetic units in spoken language and that the phonetic

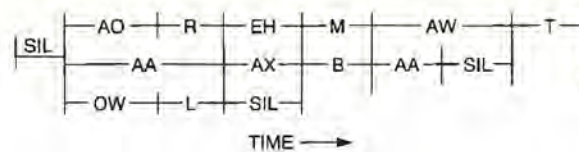


Figure 2.31 Phoneme lattice for word string.

units are broadly characterized by a set of properties that are manifest in the speech signal, or its spectrum, over time. Even though the acoustic properties of phonetic units are highly variable, both with speakers and with neighboring phonetic units (the so-called coarticulation of sounds), it is assumed that the rules governing the variability are straightforward and can readily be learned and applied in practical situations. Hence the first step in the acoustic-phonetic approach to speech recognition is called a segmentation and labeling phase because it involves segmenting the speech signal into discrete (in time) regions where the acoustic properties of the signal are representative of one (or possibly several) phonetic units (or classes), and then attaching one or more phonetic labels to each segmented region according to the acoustic properties. To actually do speech recognition, a second step is required. This second step attempts to determine a valid word (or string of words) from the sequence of phonetic labels produced in the first step, which is consistent with the constraints of the speech-recognition task (i.e., the words are drawn from a given vocabulary, the word sequence makes syntactic sense and has semantic meaning, etc.).

To illustrate the steps involved in the acoustic-phonetic approach to speech recognition, consider the phoneme lattice shown in Figure 2.31. (A phoneme lattice is the result of the segmentation and labeling step of the recognition process and represents a sequential set of phonemes that are likely matches to the spoken input speech.) The problem is to decode the phoneme lattice into a word string (one or more words) such that every instant of time is included in one of the phonemes in the lattice, and such that the word (or word sequence) is valid according to rules of English syntax. (The symbol SIL stands for silence or a pause between sounds or words; the vertical position in the lattice, at any time, is a measure of the goodness of the acoustic match to the phonetic unit, with the highest unit having the best match.) With a modest amount of searching, one can derive the appropriate phonetic string SIL-AO-L-AX-B-AW-T corresponding to the word string “all about,” with the phonemes L, AX, and B having been second or third choices in the lattice and all other phonemes having been first choices. This simple example illustrates well the difficulty in decoding phonetic units into word strings. This is the so-called lexical access problem. Interestingly, as we will see in the next section, the real problem with the acoustic-phonetic approach to speech recognition is the difficulty in getting a reliable phoneme lattice for the lexical access stage.

The pattern-recognition approach to speech recognition is basically one in which the speech patterns are used directly without explicit feature determination (in the acoustic-phonetic sense) and segmentation. As in most pattern-recognition approaches, the method has two steps—namely, training of speech patterns, and recognition of patterns via pattern comparison. Speech “knowledge” is brought into the system via the training procedure. The concept is that if enough versions of a pattern to be recognized (be it a sound, a word, a phrase, etc.) are included in a training set provided to the algorithm, the training procedure

should be able to adequately characterize the acoustic properties of the pattern (with no regard for or knowledge of any other pattern presented to the training procedure). This type of characterization of speech via training is called pattern classification because the machine learns which acoustic properties of the speech class are reliable and repeatable across all training tokens of the pattern. The utility of the method is the pattern-comparison stage, which does a direct comparison of the unknown speech (the speech to be recognized), with each possible pattern learned in the training phase and classifies the unknown speech according to the goodness of match of the patterns.

The pattern-recognition approach to speech recognition is the basis for the remainder of this book. Hence there will be a great deal of discussion and explanation of virtually every aspect of the procedure. However, at this point, suffice it to say that the pattern-recognition approach is the method of choice for speech recognition for three reasons:

1. **Simplicity of use.** The method is easy to understand, it is rich in mathematical and communication theory justification for individual procedures used in training and decoding, and it is widely used and understood.
2. **Robustness and invariance to different speech vocabularies, users, feature sets, pattern comparison algorithms and decision rules.** This property makes the algorithm appropriate for a wide range of speech units (ranging from phonemelike units all the way through words, phrases, and sentences), word vocabularies, talker populations, background environments, transmission conditions, etc.
3. **Proven high performance.** It will be shown that the pattern-recognition approach to speech recognition consistently provides high performance on any task that is reasonable for the technology and provides a clear path for extending the technology in a wide range of directions such that the performance degrades gracefully as the problem becomes more and more difficult.

The so-called artificial intelligence approach to speech recognition is a hybrid of the acoustic-phonetic approach and the pattern-recognition approach in that it exploits ideas and concepts of both methods. The artificial intelligence approach attempts to mechanize the recognition procedure according to the way a person applies its intelligence in visualizing, analyzing, and finally making a decision on the measured acoustic features. In particular, among the techniques used within this class of methods are the use of an expert system for segmentation and labeling so that this crucial and most difficult step can be performed with more than just the acoustic information used by pure acoustic-phonetic methods (in particular, methods that integrate phonemic, lexical, syntactic, semantic, and even pragmatic knowledge into the expert system have been proposed and studied); learning and adapting over time (i.e., the concept that knowledge is often both static and dynamic and that models must adapt to the dynamic component of the data); the use of neural networks for learning the relationships between phonetic events and all known inputs (including acoustic, lexical, syntactic, semantic, etc.) as well as for discrimination between similar sound classes.

The use of neural networks could represent a separate structural approach to speech recognition or be regarded as an implementational architecture that may be incorporated

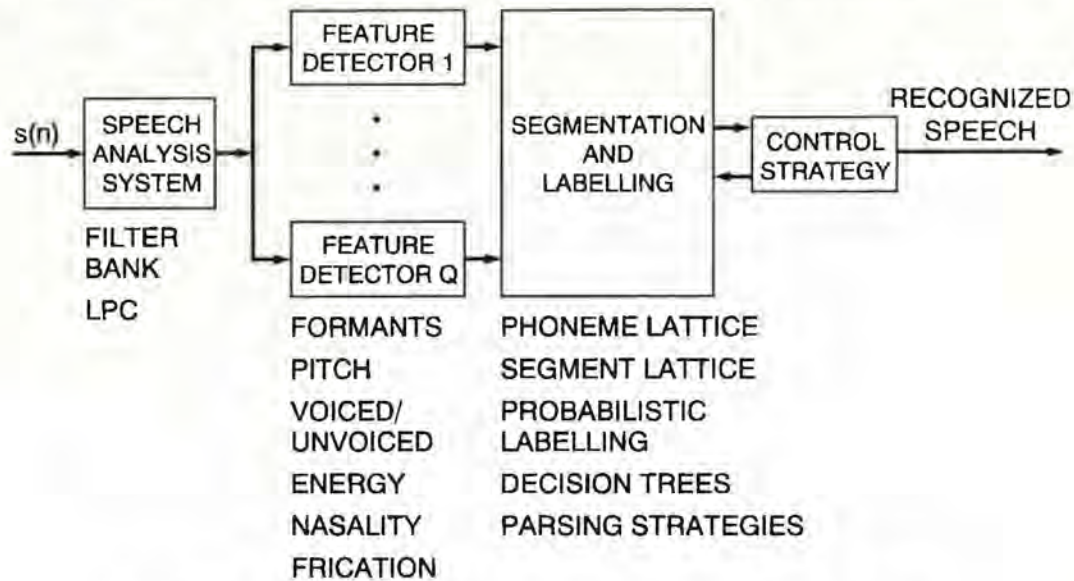


Figure 2.32 Block diagram of acoustic-phonetic speech-recognition system.

in any of the above three classical approaches. The concepts and ideas of applying neural networks to speech-recognition problems are relatively new; hence we will devote a fair amount of discussion within this chapter to outline the basic ways in which neural networks are used in general, and applied to problems in speech recognition, in particular. In the next several sections we expand on the ideas of these three general approaches to speech recognition by machine.

2.5.1 Acoustic-Phonetic Approach to Speech Recognition

Figure 2.32 shows a block diagram of the acoustic-phonetic approach to speech recognition. The first step in the processing (a step common to all approaches to speech recognition) is the speech analysis system (the so-called feature measurement method), which provides an appropriate (spectral) representation of the characteristics of the time-varying speech signal. The most common techniques of spectral analysis are the class of filter bank methods and the class of linear predictive coding (LPC) methods. The properties of these methods will be discussed in great detail in Chapter 3. Broadly speaking, both of these methods provide spectral descriptions of the speech over time.

The next step in the processing is the feature-detection stage. The idea here is to convert the spectral measurements to a set of features that describe the broad acoustic properties of the different phonetic units. Among the features proposed for recognition are nasality (presence or absence of nasal resonance), frication (presence or absence of random excitation in the speech), formant locations (frequencies of the first three resonances), voiced-unvoiced classification (periodic or aperiodic excitation), and ratios of high- and low-frequency energy. Many proposed features are inherently binary (e.g., nasality, frication, voiced-unvoiced); others are continuous (e.g., formant locations, energy ratios). The feature-detection stage usually consists of a set of detectors that operate in parallel and use appropriate processing and logic to make the decision as to presence or absence, or

value, of a feature. The algorithms used for individual feature detectors are sometimes sophisticated ones that do a lot of signal processing, and sometimes they are rather trivial estimation procedures.

The third step in the procedure is the segmentation and labeling phase whereby the system tries to find stable regions (where the features change very little over the region) and then to label the segmented region according to how well the features within that region match those of individual phonetic units. This stage is the heart of the acoustic-phonetic recognizer and is the most difficult one to carry out reliably; hence various control strategies are used to limit the range of segmentation points and label possibilities. For example, for individual word recognition, the constraint that a word contains at least two phonetic units and no more than six phonetic units means that the control strategy need consider solutions with between 1 and 5 internal segmentation points. Furthermore, the labeling strategy can exploit lexical constraints on words to consider only words with n phonetic units whenever the segmentation gives $n - 1$ segmentation points. These constraints are often powerful ones that reduce the search space and significantly increase performance (accuracy of segmentation and labeling) of the system.

The result of the segmentation and labeling step is usually a phoneme lattice (of the type shown in Figure 2.31) from which a lexical access procedure determines the best matching word or sequence of words. Other types of lattices (e.g., syllable, word) can also be derived by integrating vocabulary and syntax constraints into the control strategy as discussed above. The quality of the matching of the features, within a segment, to phonetic units can be used to assign probabilities to the labels, which then can be used in a probabilistic lexical access procedure. The final output of the recognizer is the word or word sequence that best matches, in some well-defined sense, the sequence of phonetic units in the phoneme lattice.

2.5.1.1 Acoustic Phonetic Vowel Classifier

To illustrate the labeling procedure on a segment classified as a vowel, consider the flow chart of Figure 2.33. We assume that three features have been detected over the segment—namely, first formant, F_1 , second formant, F_2 , and duration of the segment, D . Consider just the set of steady vowels (i.e., we exclude the diphthongs). To classify a vowel segment as one of the 10 steady vowels, several tests can be made to separate groups of vowels. As shown in Figure 2.33 the first test separates vowels with low F_1 (called diffuse vowels and including /i/, /I/, /ə/, /U/, /u/) from vowels with high F_1 (called compact vowels and including /ε/, /æ/, /a/, /Λ/, /ɔ/). Each of these subsets can be split further on the basis of F_2 measurements, with acute vowels having high F_2 and grave vowels having low F_2 . The third test is one based on segment duration, which separates tense vowels (large values of D) from lax vowels (small values of D). Finally, a finer test on formant values separates the remaining unresolved vowels, resolving the vowels into flat vowels (where $F_1 + F_2$ exceeds a threshold T) and plain vowels (where $F_1 + F_2$ falls below the threshold T).

It should be clear that there are several thresholds embedded within the vowel classifier. Such thresholds are often determined experimentally so as to maximize classification accuracy on a given corpus of speech.

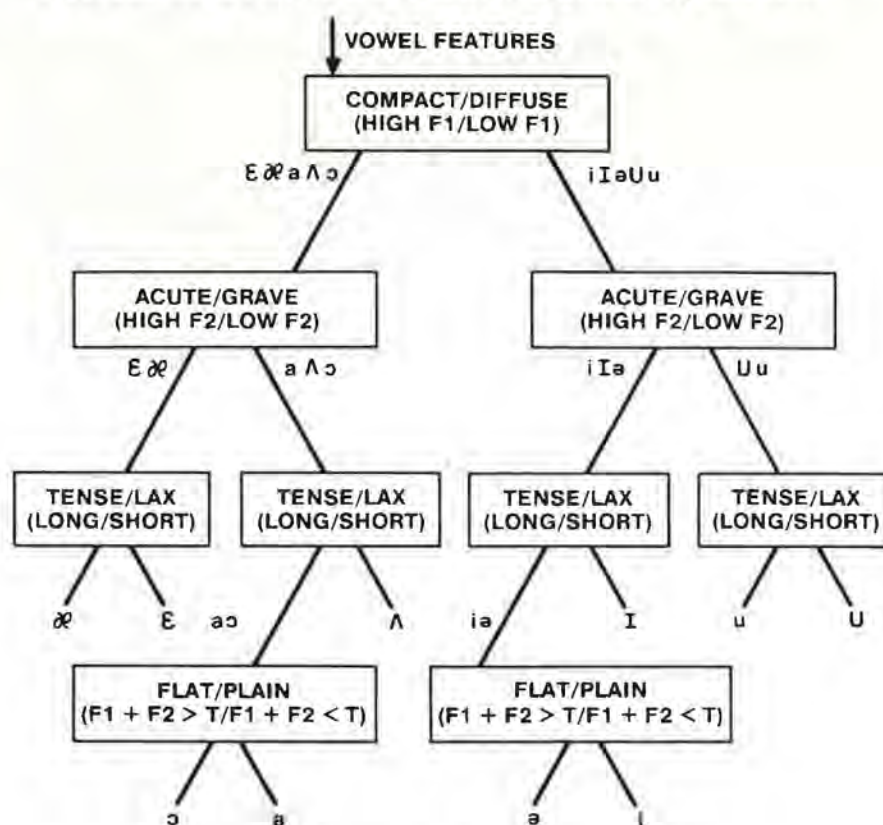


Figure 2.33 Acoustic-phonetic vowel classifier.

2.5.1.2 Speech Sound Classifier

Vowel classification is just a small part of the phonetic labeling procedure of an acoustic-phonetic recognizer. In theory, one needs a method of classifying an arbitrary segment into one (or more) of the 40 plus phonetic units discussed earlier in this chapter. Rather than discussing how to solve this very difficult problem, consider the somewhat simpler problem of classifying a speech segment into one of several broad speech classes—e.g., unvoiced stop, voiced stop, unvoiced fricative. Again there is no simple or generally well-accepted procedure for accomplishing this task; however, we show in Figure 2.34 one simple and straightforward way to accomplish such a classification.

The method uses a binary tree to make decisions as to various broad sound classes. The first decision is a *sound/silence* split in which the speech features (primarily energy in this case) are compared to selected thresholds, and *silence* is split off if the test is negative for speech sounds. The second decision is a *voiced/unvoiced* decision (primarily based on the presence of periodicity within the segment) in which unvoiced sounds are split apart from voiced sounds. A test for unvoiced stop consonants is made (seeing if a stop gap of silence preceded the segment), and this separates the unvoiced stops (/t/, /p/, /k/, /ç/) from the unvoiced fricatives (/f/, /θ/, /s/, /ʃ/). A *high-frequency/low-frequency* (energy) test separates voiced fricatives (/v/, /ð/, /z/, /ʒ/) from other voiced sounds. Voiced stops are separated out by checking to see whether the preceding sound is silence (or silencelike). Finally, a *vowel/sonorant* spectral test (searching for spectral gaps) separates vowels from

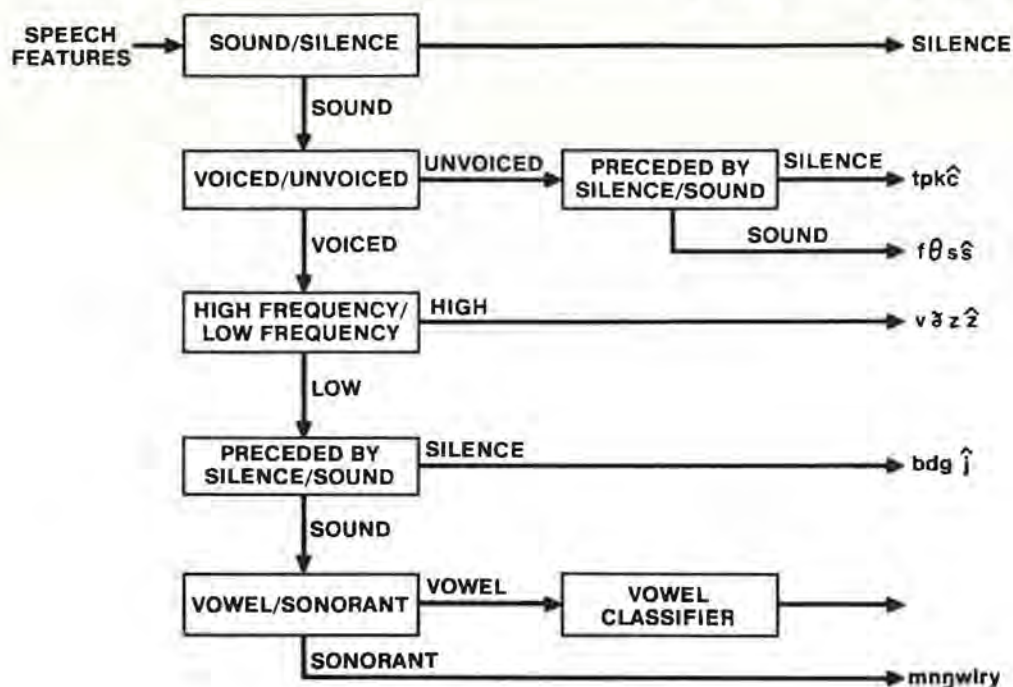


Figure 2.34 Binary tree speech sound classifier.

sonorants (nasal consonants and /w/, /l/, /r/, and /y/). The vowel classifier of Figure 2.33 can then be used for finer vowel distinctions.

The tests shown in Figure 2.34 are rather crude and are therefore highly prone to error. For example, some voiced stop consonants are *not* preceded by silence or by a silencelike sound. Another problem is that no way of distinguishing diphthongs from vowels is provided. Virtually every decision in the binary tree is subject to scrutiny as to its utility in any practical system.

2.5.1.3 Examples of Acoustic Phonetic Labeling

To illustrate some of the difficulties faced by the acoustic-phonetic approach to speech recognition, consider the following example. (Shown in the example is the phonetic labeling of a sentence [only the top-choice phonetic candidate is shown for each segment], along with its decoding into the proper word sequence.) In this example (taken from an actual acoustic-phonetic recognizer) we see that there are inserted phonetic units (Y in “MAY,” AX in “BY”), deleted phonetic units (N in “EARN,” N in “MONEY”), and phonetic substitutions (J for K in “WORKING,” N for NG in “WORKING”). The difficulty of proper decoding of phonetic units into words and sentences grows dramatically with increases in the rates of phoneme insertion, deletion, and substitution.

phonemes:	/sil/	-/j/-/e/-/n/	/m/-/e/-/y/	/ɜ/-/m/-/ɔ/-/r/	/m/-/ɹ/-/sil/-/e/
ARPABET:	SIL	-JH-EY-N + M-EY-Y + ER-M-AO-R + M-AH-SIL-EY			
words:	JANE	MAY	EARN MORE	MONEY	

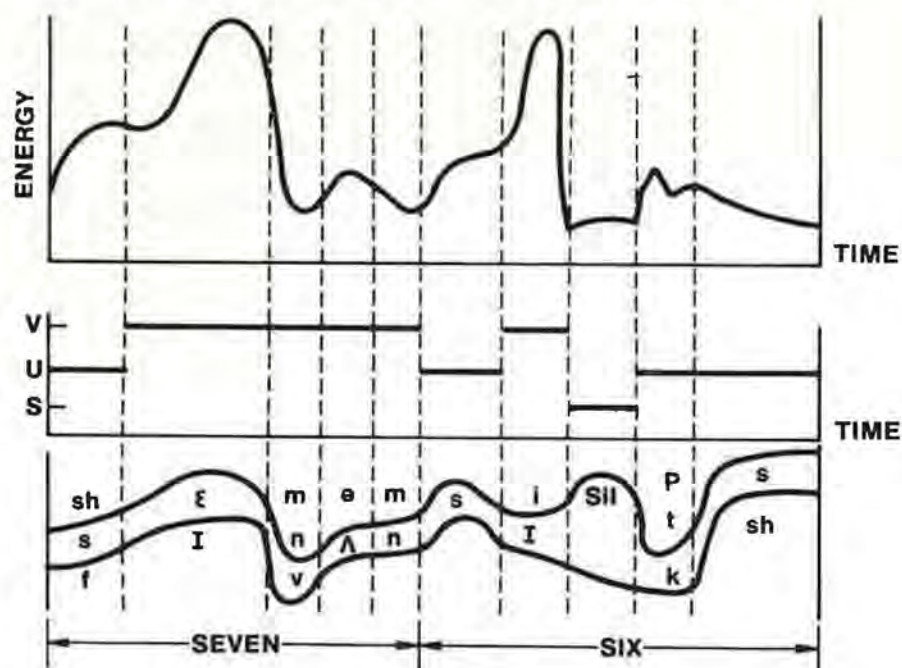


Figure 2.35 Segmentation and labeling for word sequence "seven-six."

phonemes: /b/-/aʔ/-/ə/ /w/-/ɜ/-/sɪl/-/j/-/l/-/n/ /h/-/a/-/r/-/sɪl/-/d/
 ARPABET: B -AY-AX+ W-ER-SIL- J -IH- N +HH-AA- R-SIL- D
 words: BY WORKING HARD

Two other examples of acoustic-phonetic segmentation and labeling are given in Figures 2.35 and 2.36. Shown in these figures are the energy contour of the speech signal, the voiced-unvoiced-silence classification over time, the segmentation points, and the lattice of phonetic units. The "proper" decoding of the lattice corresponding to the spoken word is shown as the phonetic units enclosed within the solid heavy lines. For the example of Figure 2.35 (the digit sequence "seven-six"), we see that although most top phoneme candidate errors are within the same sound class (e.g., /sh/ instead of /s/), some errors are between classes (e.g., /m/ instead of /v/). For decoding into digits, such cross-class errors are usually of little significance.

For the example of Figure 2.36 (the word sequence "did you"), the decoding into phonetic units is only the first step in a difficult decoding problem, because the basic speech sounds of the words "did" and "you" are phonologically changed in context from D-IH-D-Y-UW to D-IH-J-UH. This phonological effect exacerbates the problem of acoustic phonetic decoding even further than the insertion/deletion/substitution problems mentioned earlier.

2.5.1.4 Issues in Acoustic Phonetic Approach

Many problems are associated with the acoustic-phonetic approach to speech recognition. These problems, in many ways, account for the lack of success in practical speech-recognition systems. Among these are the following:

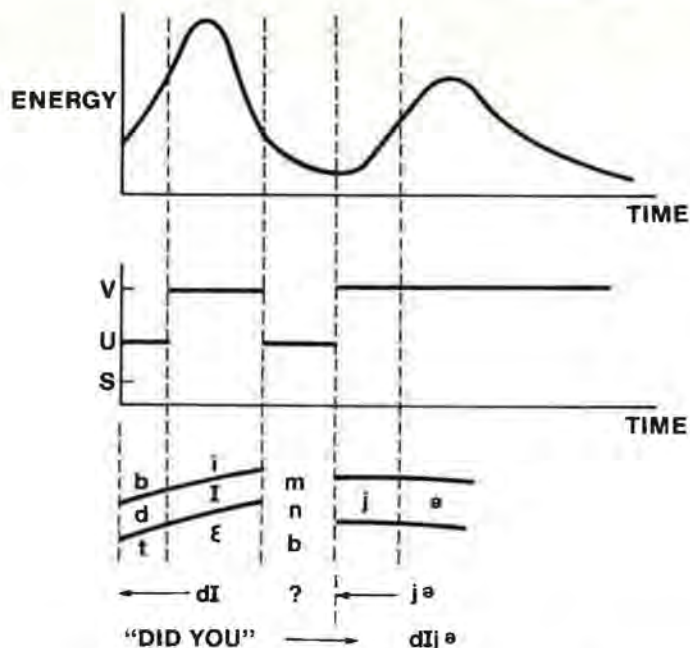


Figure 2.36 Segmentation and labeling for word sequence "did you."

1. The method requires extensive knowledge of the acoustic properties of phonetic units. (Recall that the existence of phonetic units is assumed a priori in the acoustic-phonetic approach. Knowledge of acoustic properties of these phonetic units often is established in an a posteriori manner.) This knowledge is, at best incomplete, and at worst totally unavailable for all but the simplest of situations (e.g., steady vowels).
2. The choice of features is made mostly based on ad hoc considerations. For most systems the choice of features is based on intuition and is not optimal in a well-defined and meaningful sense.
3. The design of sound classifiers is also not optimal. Ad hoc methods are generally used to construct binary decision trees. More recently classification and regression tree (CART) methods have been used to make the decision trees more robust [10]. However, since the choice of features is most likely to be suboptimal, optimal implementation of CART is rarely achieved.
4. No well-defined, automatic procedure exists for tuning the method (i.e., adjusting decision thresholds, etc.) on real, labeled speech. In fact, there is not even an ideal way of labeling the training speech in a manner consistent and agreed on uniformly by a wide class of linguistic experts.

Because of all these problems, the acoustic-phonetic method of speech recognition remains an interesting idea but one that needs much more research and understanding before it can be used successfully in actual speech-recognition problems.

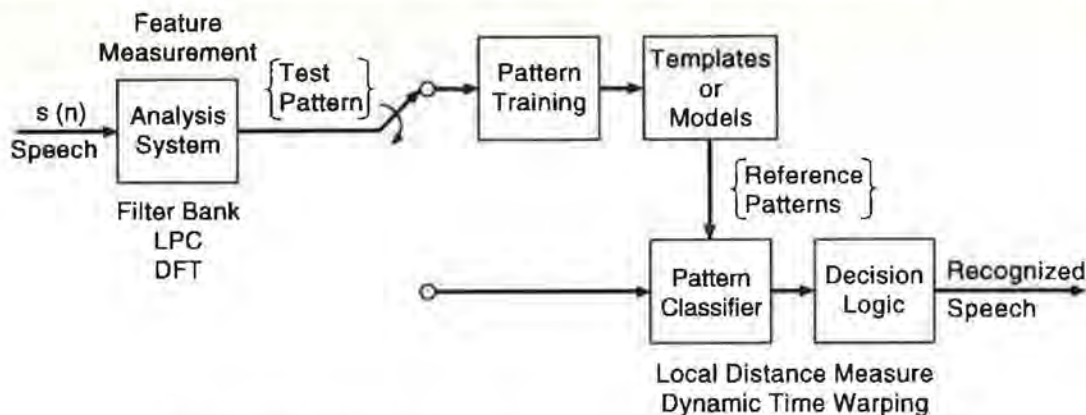


Figure 2.37 Block diagram of pattern-recognition speech recognizer.

2.5.2 Statistical Pattern-Recognition Approach to Speech Recognition

A block diagram of a canonic pattern-recognition approach to speech recognition is shown in Figure 2.37. The pattern-recognition paradigm has four steps, namely:

1. Feature measurement, in which a sequence of measurements is made on the input signal to define the “test pattern.” For speech signals the feature measurements are usually the output of some type of spectral analysis technique, such as a filter bank analyzer, a linear predictive coding analysis, or a discrete Fourier transform (DFT) analysis.
2. Pattern training, in which one or more test patterns corresponding to speech sounds of the same class are used to create a pattern representative of the features of that class. The resulting pattern, generally called a reference pattern, can be an exemplar or template, derived from some type of averaging technique, or it can be a model that characterizes the statistics of the features of the reference pattern.
3. Pattern classification, in which the unknown test pattern is compared with each (sound) class reference pattern and a measure of similarity (distance) between the test pattern and each reference pattern is computed. To compare speech patterns (which consist of a sequence of spectral vectors), we require both a local distance measure, in which local distance is defined as the spectral “distance” between two well-defined spectral vectors, and a global time alignment procedure (often called a dynamic time warping algorithm), which compensates for different rates of speaking (time scales) of the two patterns.
4. Decision logic, in which the reference pattern similarity scores are used to decide which reference pattern (or possibly which sequence of reference patterns) best matches the unknown test pattern.

The factors that distinguish different pattern-recognition approaches are the types of feature

measurement, the choice of templates or models for reference patterns, and the method used to create reference patterns and classify unknown test patterns.

The remaining chapters of this book will discuss all aspects of the model shown in Figure 2.37. The general strengths and weaknesses of the pattern recognition model include the following:

1. The performance of the system is sensitive to the amount of training data available for creating sound class reference patterns; generally the more training, the higher the performance of the system for virtually any task.
2. The reference patterns are sensitive to the speaking environment and transmission characteristics of the medium used to create the speech; this is because the speech spectral characteristics are affected by transmission and background noise.
3. No speech-specific knowledge is used explicitly in the system; hence, the method is relatively insensitive to choice of vocabulary words, task, syntax, and task semantics.
4. The computational load for both pattern training and pattern classification is generally linearly proportional to the number of patterns being trained or recognized; hence, computation for a large number of sound classes could and often does become prohibitive.
5. Because the system is insensitive to sound class, the basic techniques are applicable to a wide range of speech sounds, including phrases, whole words, and subword units. Hence we will see how a basic set of techniques developed for one sound class (e.g., words) can generally be directly applied to different sound classes (e.g., subword units) with little or no modifications to the algorithms.
6. It is relatively straightforward to incorporate syntactic (and even semantic) constraints directly into the pattern-recognition structure, thereby improving recognition accuracy and reducing computation.

2.5.3 Artificial Intelligence (AI) Approaches to Speech Recognition

The basic idea of the artificial intelligence approach to speech recognition is to compile and incorporate knowledge from a variety of knowledge sources and to bring it to bear on the problem at hand. Thus, for example, the AI approach to segmentation and labeling would be to augment the generally used acoustic knowledge with phonemic knowledge, lexical knowledge, syntactic knowledge, semantic knowledge, and even pragmatic knowledge. To be more specific, we first define these different knowledge sources:

- acoustic knowledge—evidence of which sounds (predefined phonetic units) are spoken on the basis of spectral measurements and presence or absence of features
- lexical knowledge—the combination of acoustic evidence so as to postulate words as specified by a lexicon that maps sounds into words (or equivalently decomposes words into sounds)
- syntactic knowledge—the combination of words to form grammatically correct strings (according to a language model) such as sentences or phrases

- semantic knowledge—understanding of the task domain so as to be able to validate sentences (or phrases) that are consistent with the task being performed, or which are consistent with previously decoded sentences
- pragmatic knowledge—inference ability necessary in resolving ambiguity of meaning based on ways in which words are generally used.

To illustrate the correcting and constraining power of these knowledge sources, consider the following sentences:

1. Go to the refrigerator and get me a book.
2. The bears killed the rams.
3. Power plants colorless happily old.
4. Good ideas often run when least expected.

The first sentence is syntactically meaningful but semantically inconsistent. The second sentence can be interpreted in at least two pragmatically different ways, depending on whether the context is an event in a jungle or the description of a football game between two teams called the “bears” and the “rams.” The third sentence is syntactically unacceptable and semantically meaningless. The fourth sentence is semantically inconsistent and can trivially be corrected by changing the word *run* to *come*, a slight phonetic difference.

The word-correcting capability of higher-level knowledge sources is illustrated in Figure 2.38, which shows the word error probability of a recognizer both with and without syntactic constraints, as a function of a “deviation” parameter σ . As the deviation parameter gets larger, the word error probability increases for both cases; however, without syntax the word error probability rapidly leads to 1.0, but with syntax it increases gradually with increases in the noise parameter.

There are several ways to integrate knowledge sources within a speech recognizer. Perhaps the most standard approach is the “bottom-up” processor (Figure 2.39), in which the lowest-level processes (e.g., feature detection, phonetic decoding) precede higher-level processes (lexical decoding, language model) in a sequential manner so as to constrain each stage of the processing as little as possible. An alternative is the so-called “top-down” processor, in which the language model generates word hypotheses that are matched against the speech signal, and syntactically and semantically meaningful sentences are built up on the basis of the word match scores. Figure 2.40 shows a system that is often implemented in the top-down mode by integrating the unit matching, lexical decoding, and syntactic analyses modules into a consistent framework. (This system will be discussed extensively in the chapter on large-vocabulary continuous-speech recognition.)

A third alternative is the so-called blackboard approach, as illustrated in Figure 2.41. In this approach, all knowledge sources (KS) are considered independent; a hypothesis-and-test paradigm serves as the basic medium of communication among KSs; each KS is data driven, based on the occurrence of patterns on the blackboard that match the templates specified by the KS; the system activity operates asynchronously; assigned cost and utility considerations and an overall ratings policy to combine and propagate ratings across all levels. The blackboard approach was extensively studied at CMU in the 1970s [11].

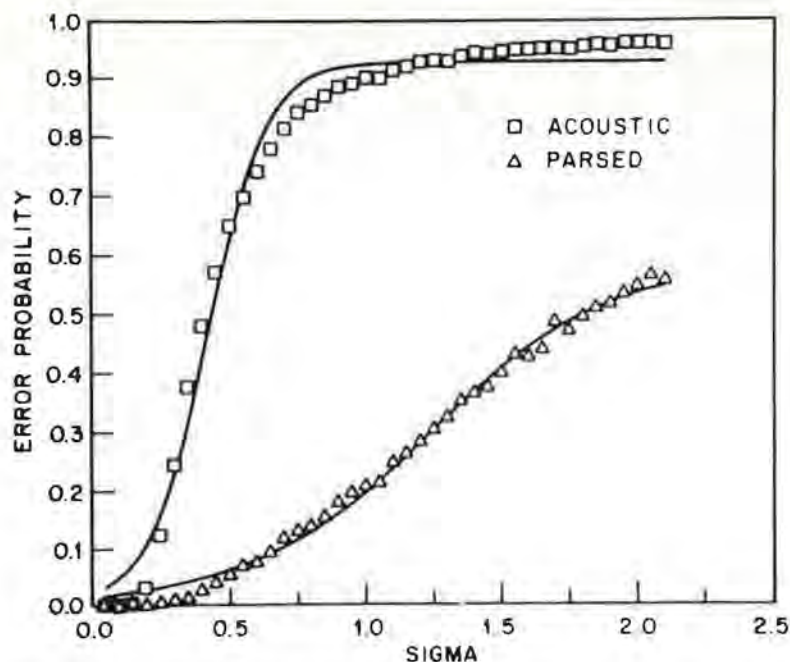


Figure 2.38 Illustration of the word correction capability of syntax in speech recognition (after Rabiner and Levinson [1]).

2.5.4 Neural Networks and Their Application to Speech Recognition

A variety of knowledge sources need to be established in the AI approach to speech recognition. Therefore, two key concepts of artificial intelligence are automatic knowledge acquisition (learning) and adaptation. One way in which these concepts have been implemented is via the neural network approach. In this section, we discuss the motivation for why people have studied neural networks and how they have been applied to speech-recognition systems.

Figure 2.42 shows a conceptual block diagram of a speech understanding system loosely based on a model of speech perception in human beings. The acoustic input signal is analyzed by an “ear model” that provides spectral information (over time) about the signal and stores it in a sensory information store. Other sensory information (e.g., from vision or touch) is available in the sensory information store and is used to provide several “feature-level” descriptions of the speech. Both long-term (static) and short-term (dynamic) memory are available to the various feature detectors. Finally, after several stages of refined feature detection, the final output of the system is an interpretation of the information in the acoustic input.

The system of Figure 2.42 is meant to model the human speech understanding system. The auditory analysis is based loosely on our understanding of the acoustic processing in the ear. The various feature analyses represent processing at various levels in the neural pathways to the brain. The short- and long-term memory provide external control of the neural processes in ways that are not well understood. The overall form of the model is that of a feed forward connectionist network—that is, a neural net. To better explain the strengths and limitations of neural networks, we now give a brief introduction to the issues

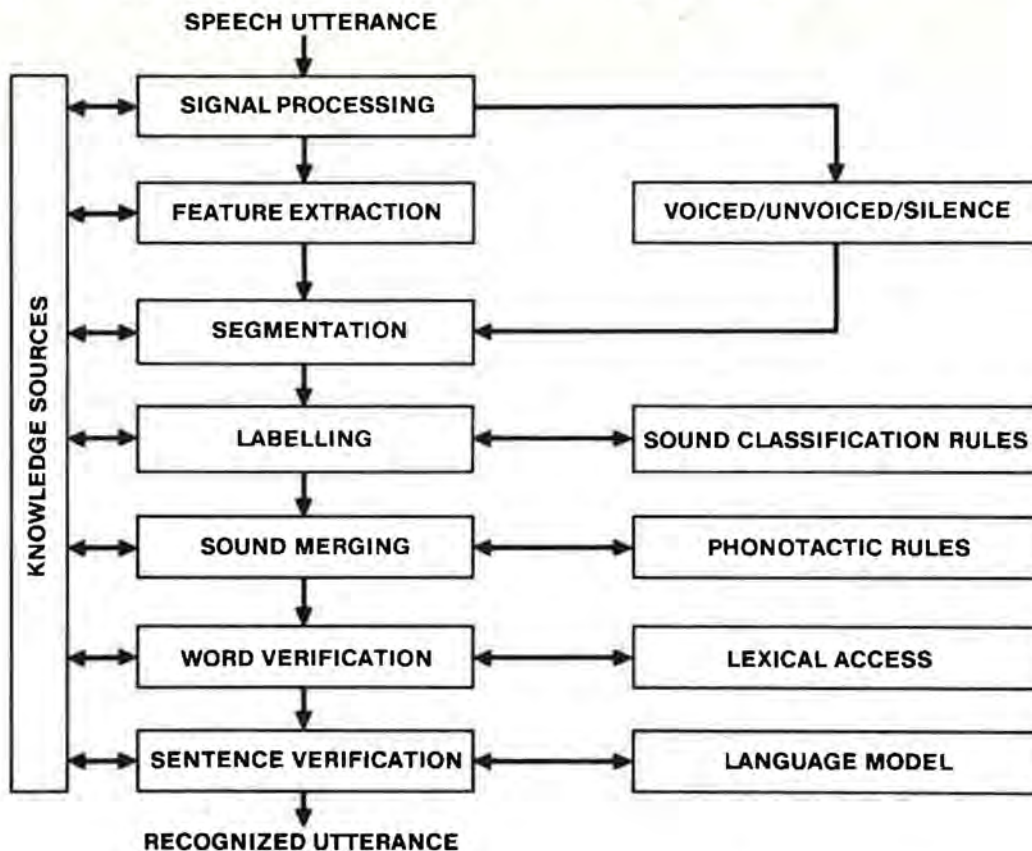


Figure 2.39 A bottom-up approach to knowledge integration for speech recognition.

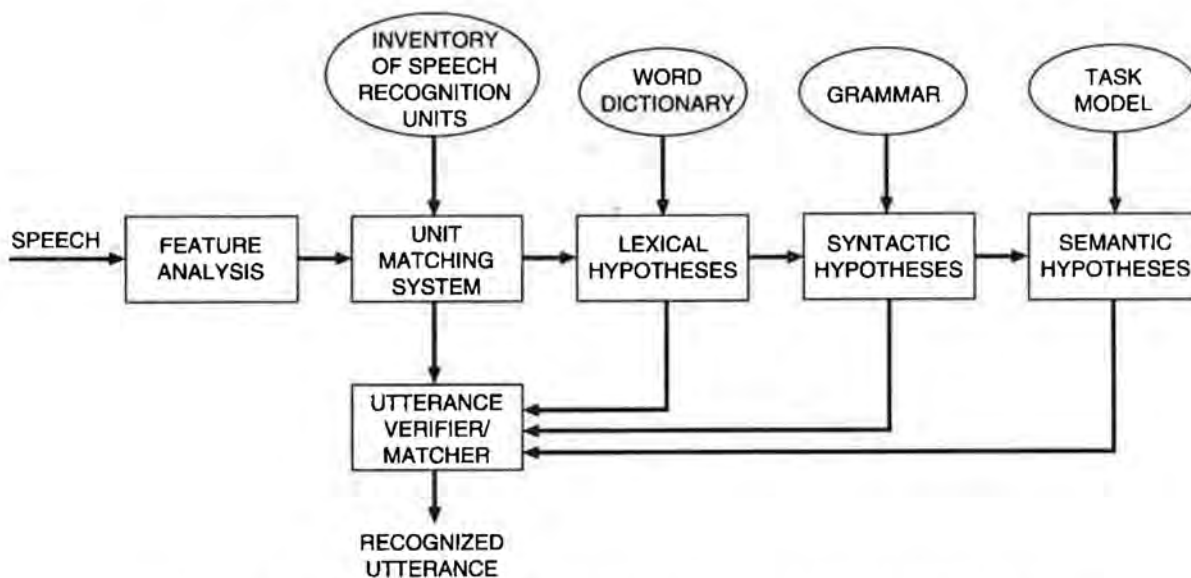


Figure 2.40 A top-down approach to knowledge integration for speech recognition.

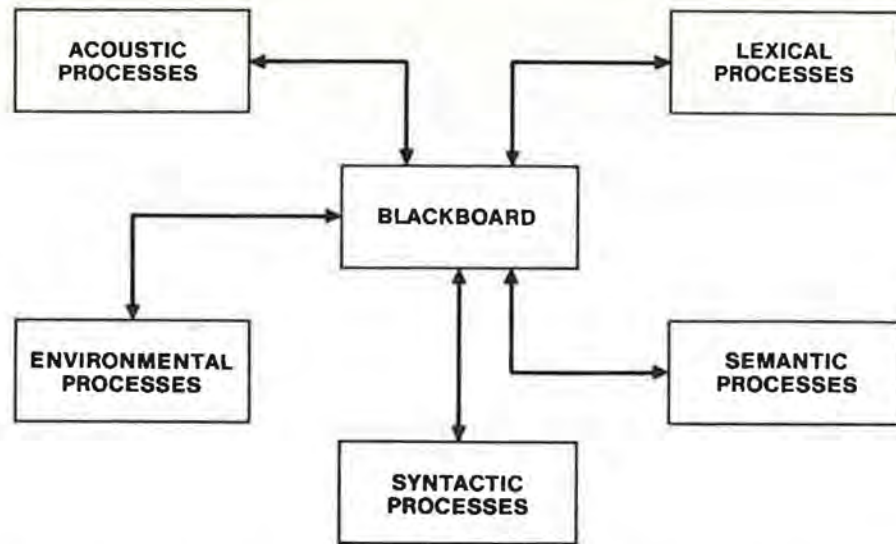


Figure 2.41 A blackboard approach to knowledge integration for speech recognition (after Lesser et al. [11]).

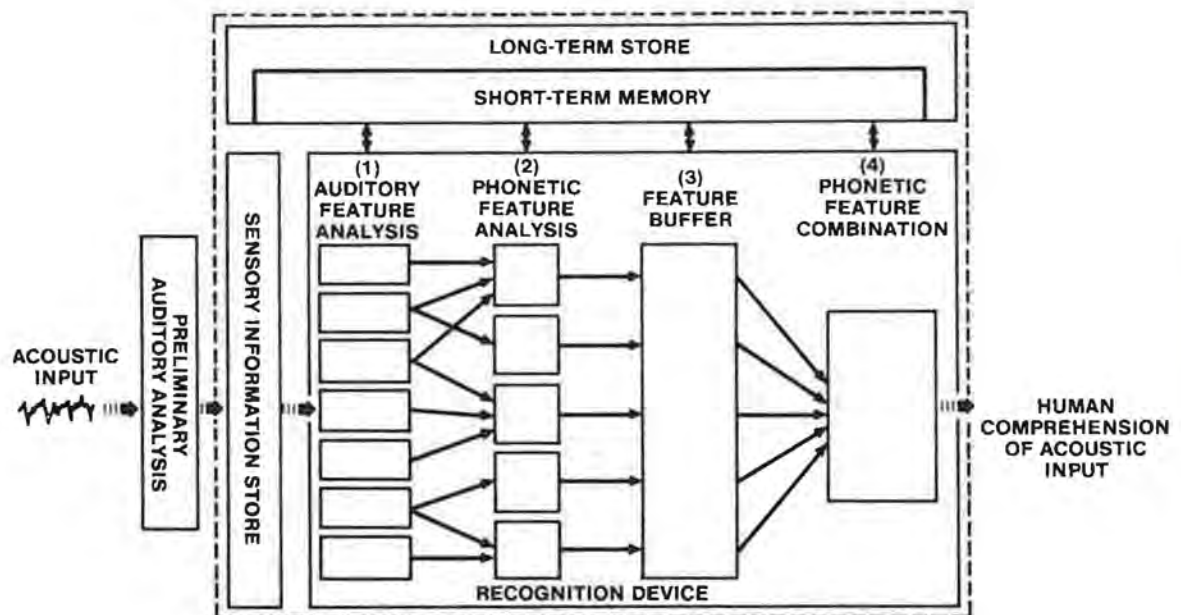


Figure 2.42 Conceptual block diagram of a human speech understanding system.

in the theory and implementations of neural networks. Then we return to some practical proposals for how neural networks could implement actual speech recognizers.

2.5.4.1 Basics of Neural Networks

A neural network, which is also called a connectionist model, a neural net, or a parallel distributed processing (PDP) model, is basically a dense interconnection of simple, non-linear, computation elements of the type shown in Figure 2.43. It is assumed that there

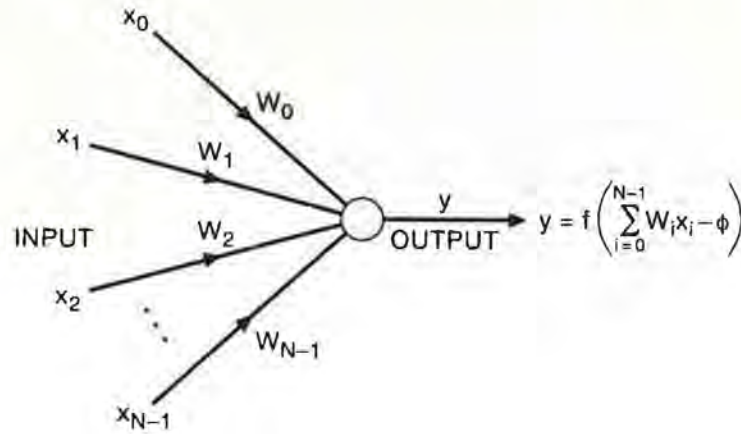


Figure 2.43 Simple computation element of a neural network.

are N inputs, labeled x_1, x_2, \dots, x_N , which are summed with weights w_1, w_2, \dots, w_N , thresholded, and then nonlinearly compressed to give the output y , defined as

$$y = f\left(\sum_{i=1}^N w_i x_i - \phi\right), \quad (2.1)$$

where ϕ is an internal threshold or offset, and f is a nonlinearity of one of the types given below:

1. hard limiter

$$f(x) = \begin{cases} +1, & x \leq 0 \\ -1, & x < 0 \end{cases} \quad (2.2)$$

or

2. sigmoid functions

$$f(x) = \tanh(\beta x), \quad \beta > 0 \quad (2.3)$$

or

$$f(x) = \frac{1}{1 + e^{-\beta x}}, \quad \beta > 0. \quad (2.4)$$

The sigmoid nonlinearities are used most often because they are continuous and differentiable.

The biological basis of the neural network is a model by McCulloch and Pitts [12] of neurons in the human nervous system, as illustrated in Figure 2.44. This model exhibits all the properties of the neural element of Figure 2.43, including excitation potential thresholds for neuron firing (below which there is little or no activity) and nonlinear amplification, which compresses strong input signals.

2.5.4.2 Neural Network Topologies

There are several issues in the design of so-called artificial neural networks (ANNs), which model various physical phenomena, where we define an ANN as an arbitrary connection of

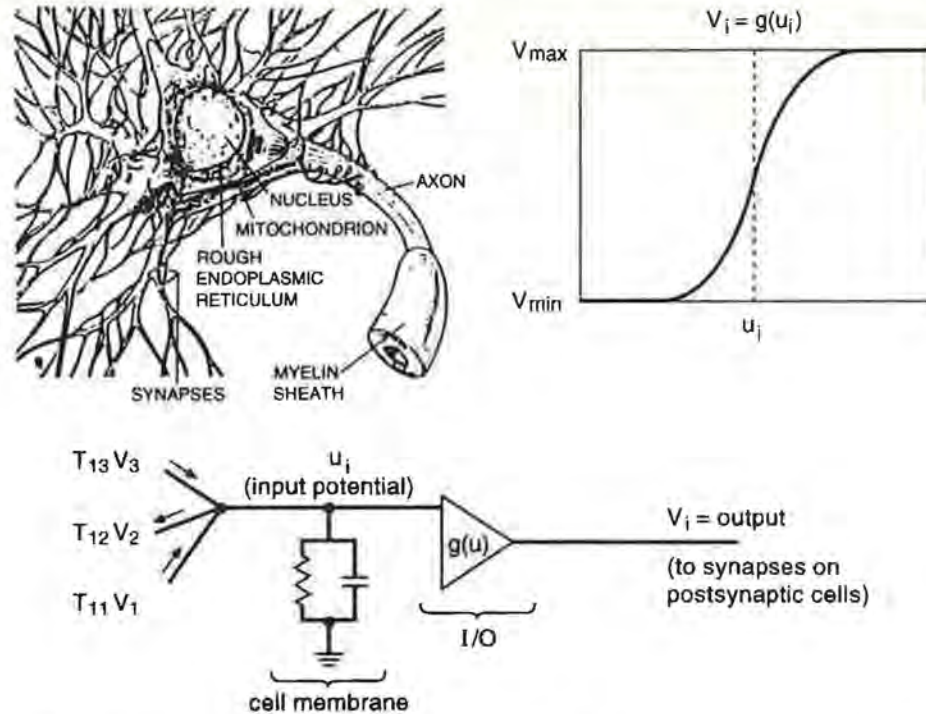


Figure 2.44 McCulloch-Pitts model of neurons (after McCulloch and Pitts [12]).

simple computational elements of the type shown in Figure 2.43. One key issue is *network topology*—that is, how the simple computational elements are interconnected. There are three standard and well known topologies:

- single/multilayer perceptrons
- Hopfield or recurrent networks
- Kohonen or self-organizing networks

In the single/multilayer perceptron, the outputs of one or more simple computational elements at one layer form the inputs to a new set of simple computational elements of the next layer. Figure 2.45 shows a single-layer perceptron and a three-layer perceptron. The single-layer perceptron has N inputs connected to M outputs in the output layer. The three-layer perceptron has two hidden layers between the input and output layers. What distinguishes the layers of the multilayer perceptron is the nonlinearity at each layer that enables the mapping between the input and output variables to possess certain particular classification/discrimination properties. For example, it can be proven that a single-layer perceptron, of the type shown in Figure 2.45a, can separate static patterns into classes with class boundaries characterized by hyperplanes in the (x_1, x_2, \dots, x_N) space. Similarly, a multilayer perceptron, with at least one hidden layer, can realize an arbitrary set of decision regions in the (x_1, \dots, x_N) space. Thus, for example, if the inputs to a multilayer perceptron are the first two speech resonances (F_1 and F_2), the network can implement a set of decision regions that partition the $(F_1 - F_2)$ space into the 10 steady state vowels, as shown in Figure 2.46 [13].

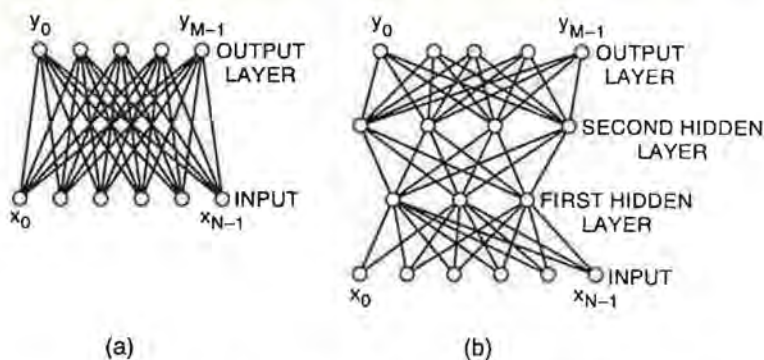
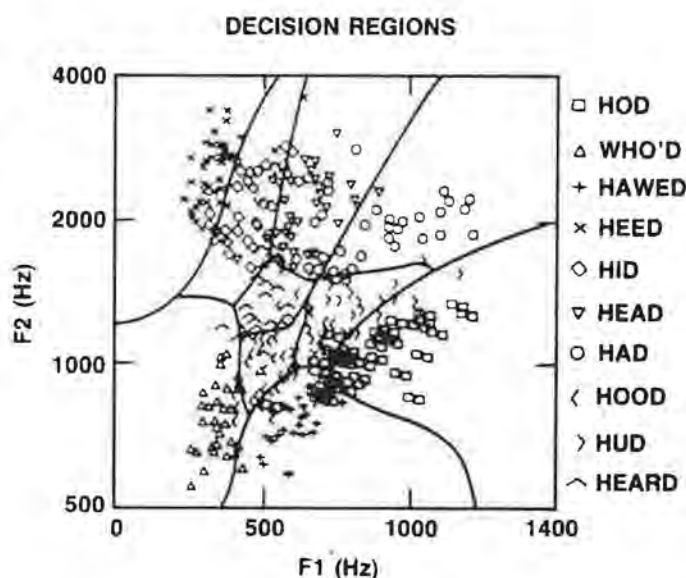
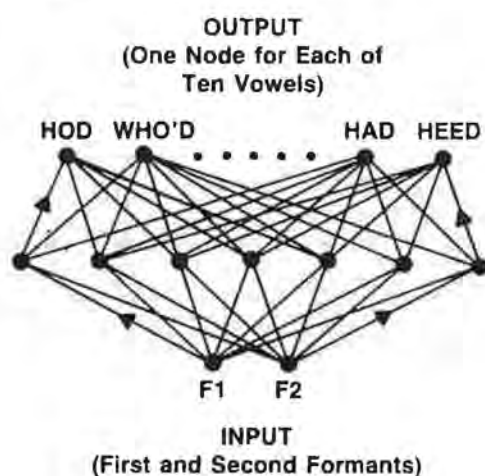


Figure 2.45 Single-layer and three-layer perceptrons.

● Perceptrons:

Figure 2.46 A multilayer perceptron for classifying steady vowels based on F_1 , F_2 measurements (after Lippmann [13]).

The Hopfield network is a *recurrent network* in which the input to each computational element includes both inputs as well as outputs. Thus with the input and output indexed by time, $x_i(t)$ and $y_i(t)$, and the weight connecting the i^{th} node and the j^{th} node denoted by w_{ij} , the basic equation for the i^{th} recurrent computational element is

$$y_i(t) = f \left[x_i(t) + \sum_j w_{ij} y_j(t-1) - \phi \right] \quad (2.5)$$

and a recurrent network with N inputs and N outputs would have the form shown in Figure 2.47. The most important property of the Hopfield network is that when $w_{ij} = w_{ji}$ and when the recurrent computation (Eq. (2.5)) is performed asynchronously, for an arbitrary constant input, the network will eventually settle to a fixed point where $y_i(t) = y_i(t-1)$ for all i . These fixed relaxation points represent stable configurations of the network and can be used in applications that have a fixed set of patterns to be matched (e.g., printed letters)

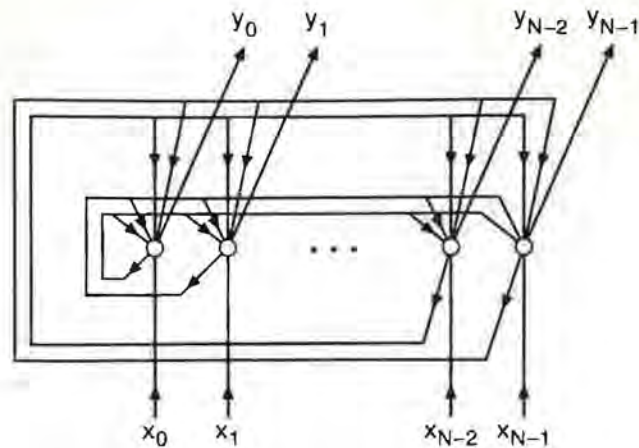


Figure 2.47 Model of a recurrent neural network.

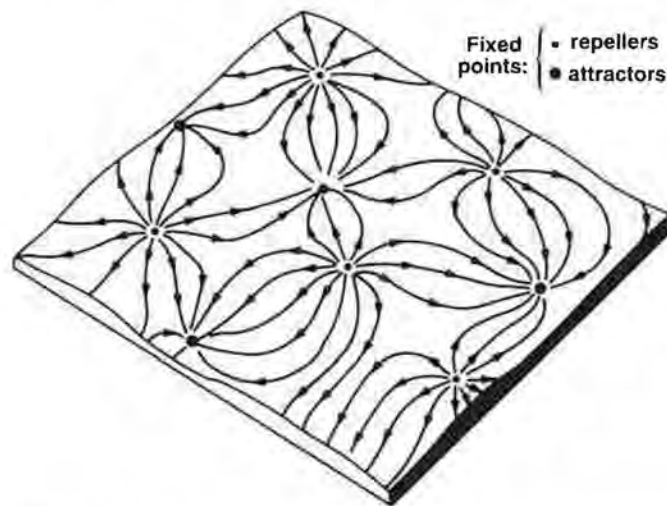


Figure 2.48 A fixed point interpretation of the Hopfield network.

in the form of a content addressable or associative memory. A simple interpretation of the Hopfield network is shown in Figure 2.48, which shows that the recurrent network has a stable set of attractors and repellers, each forming a fixed point in the input space. Every input vector, \mathbf{x} , is either “attracted” to one of the fixed points or “repelled” from another of the fixed points. The strength of this type of network is its ability to correctly classify “noisy” versions of the patterns that form the stable fixed points.

The third popular type of neural network topology is the Kohonen, self-organizing feature map, which is a clustering procedure for providing a codebook of stable patterns in the input space that characterize an arbitrary input vector, by a small number of representative clusters. We defer a discussion of this type of network to the next chapter, where the ideas of vector quantization are presented in detail.

2.5.4.3 Network Characteristics

Four model characteristics must be specified to implement an arbitrary neural network:

1. number and type of inputs—The issues involved in the choice of inputs to a neural network are similar to those involved in the choice of features for any pattern-classification system. They must provide the information required to make the decision required of the network.
2. connectivity of the network—This issue involves the size of the network—that is, the number of hidden layers and the number of nodes in each layer between input and output. There is no good rule of thumb as to how large (or small) such hidden layers must be. Intuition says that if the hidden layers are large, then it will be difficult to train the network (i.e., there will be too many parameters to estimate). Similarly, if the hidden layers are too small, the network may not be able to accurately classify all the desired input patterns. Clearly practical systems must balance these two competing effects.
3. choice of offset—The choice of the threshold, ϕ , for each computational element must be made as part of the training procedure, which chooses values for the interconnection weights (w_{ij}) and the offset, ϕ .
4. choice of nonlinearity—Experience indicates that the exact choice of the nonlinearity, f , is not very important in terms of the network performance. However, f must be continuous and differentiable for the training algorithm to be applicable.

2.5.4.4 Training of Neural Network Parameters

To completely specify a neural network, values for the weighting coefficients and the offset threshold for each computation element must be determined, based on a labeled set of training data. By a labeled training set of data, we mean an association between a set of Q input vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_Q$ and a set of Q output vectors $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_Q$ where $\mathbf{x}_1 \Rightarrow \mathbf{y}_1, \mathbf{x}_2 \Rightarrow \mathbf{y}_2, \dots, \mathbf{x}_Q \Rightarrow \mathbf{y}_Q$. For multilayer perceptrons a simple iterative, convergent procedure exists for choosing a set of parameters whose value asymptotically approaches a stationary point with a certain optimality property (e.g., a local minimum of the mean squared error, etc.). This procedure, called back propagation learning, is a simple stochastic gradient technique. For a simple, single-layer network, the training algorithm can be realized via the following convergence steps:

Perceptron Convergence Procedure

1. **Initialization:** At time $t = 0$, set $w_{ij}(0)$, ϕ_j to small random values (where w_{ij} are the weighting coefficients connecting i^{th} input node and j^{th} output node and ϕ_j is the offset to a particular computational element, and the w_{ij} are functions of time).
2. **Acquire Input:** At time t , obtain a *new* input $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$ with the desired output, $\mathbf{y}^x = \{y_1^x, y_2^x, \dots, y_M^x\}$.
3. **Calculate Output:**

$$y_j = f\left(\sum_{i=1}^N w_{ij}(t)x_i - \phi_j\right).$$

4. Adapt Weights: Update the weights as

$$w_{ij}(t+1) = w_{ij}(t) + \mathcal{T}(t) [y_j^x - y_j] \cdot x_i$$

where the “step size” $\mathcal{T}(t)$ satisfies the constraints:

a.
$$\lim_{T \rightarrow \infty} \sum_{t=1}^T \mathcal{T}(t) = \infty$$

b.
$$\lim_{T \rightarrow \infty} \sum_{t=1}^T \mathcal{T}^2(t) < \infty$$

That is, compute the gradient of the error $\sum_{j=1}^M (y_j^x - y_j)^2$ in the direction of the weight $w_{ij}(t)$. (A conventional choice of $\mathcal{T}(t)$ is $1/t$.)

5. Iteration: Iterate steps 2–4 until:

$$w_{ij}(t+1) = w_{ij}(t), \quad \forall i, t, j.$$

The perceptron convergence procedure is a slow, methodical procedure for estimating the coefficients of a system (a classifier as well as a neural network) based on a mean squared error criterion and has been extensively studied for several decades. The algorithm is simple and is guaranteed to converge, in probability, under a restricted set of conditions on $\mathcal{T}(t)$. However, its speed of convergence in many cases is not sufficiently fast. Alternative procedures for estimating neural network coefficients have been used with varying degrees of success.

2.5.4.5 Advantages of Neural Networks

Neural networks have been given serious consideration for a wide range of problems (including speech recognition) for several reasons. These include the following:

1. They can readily implement a massive degree of parallel computation. Because a neural net is a highly parallel structure of simple, identical, computational elements, it should be clear that they are prime candidates for massively parallel (analog or digital) computation.
2. They intrinsically possess a great deal of robustness or fault tolerance. Since the “information” embedded in the neural network is “spread” to every computational element within the network, this structure is inherently among the least sensitive of networks to noise or defects within the structure.
3. The connection weights of the network need not be constrained to be fixed; they can be adapted in real time to improve performance. This is the basis of the concept of adaptive learning, which is inherent in the neural network structure.
4. Because of the nonlinearity within each computational element, a sufficiently large neural network can approximate (arbitrarily closely) any nonlinearity or nonlinear dynamical system. Hence neural networks provide a convenient way of implementing nonlinear transformations between arbitrary inputs and outputs and are often more efficient than alternative physical implementations of the nonlinearity.

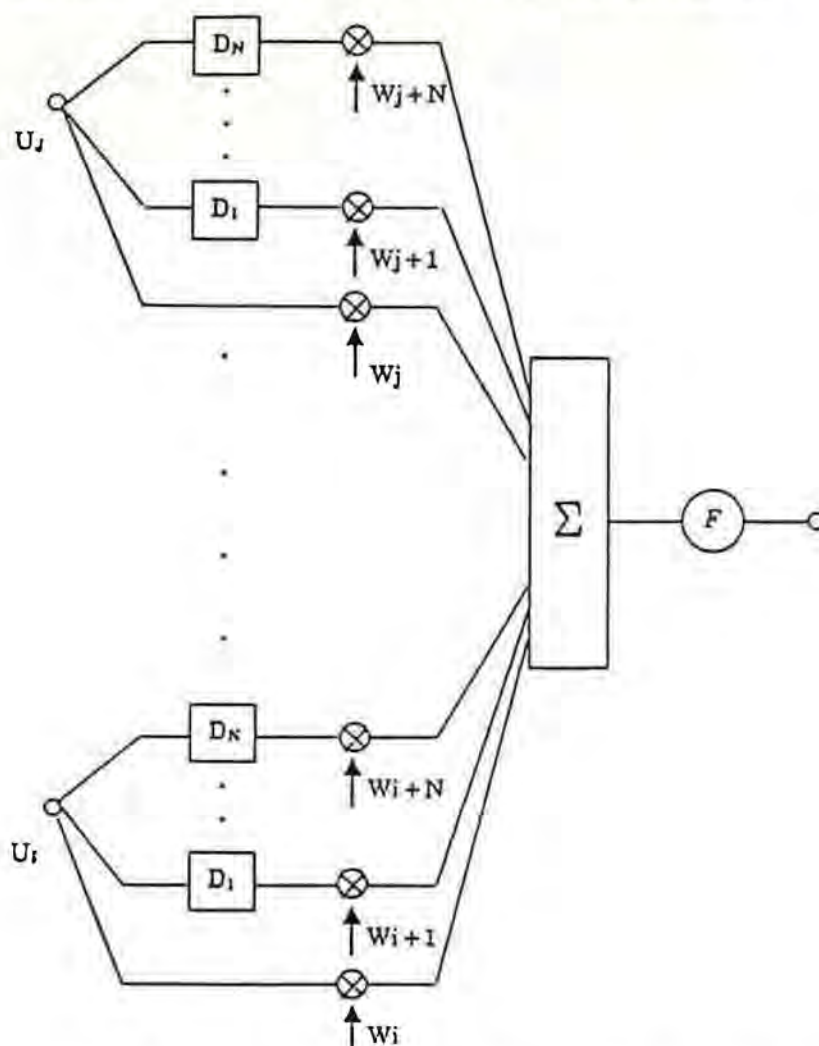


Figure 2.49 The time delay neural network computational element (after Waibel et al. [14]).

2.5.4.6 Neural Network Structures for Speech Recognition

Conventional artificial neural networks are structured to deal with static patterns. As discussed throughout this chapter, speech is inherently dynamic in nature. Hence, some modifications to the simple structures discussed in the previous sections are required for all but the simplest of problems. There is no known correct or proper way of handling speech dynamics within the framework already discussed; however, several reasonable structures have been proposed and studied and we will point out a few such structures in this section.

Perhaps the simplest neural network structure that incorporates speech pattern dynamics is the time delay neural network (TDNN) computation element shown in Figure 2.49 [14]. This structure extends the input to each computational element to include N speech frames (i.e., spectral vectors that cover a duration of $N\Delta$ seconds, where Δ is the time separation between adjacent speech spectra). By expanding the input to N frames (where N is on the order of 15), various types of acoustic-phonetic detectors become practical via

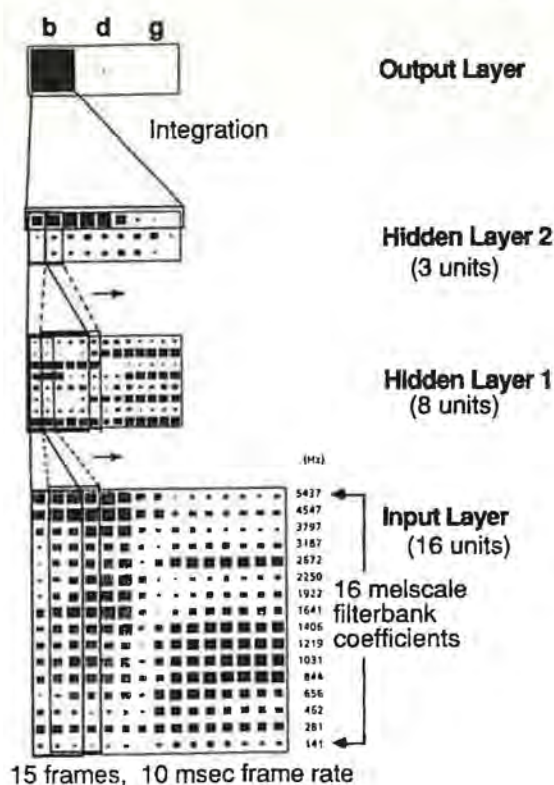


Figure 2.50 A TDNN architecture for recognizing /b/, /d/ and /g/ (after Waibel et al. [14]).

the TDNN. For example, Figure 2.50 shows a TDNN network with two hidden layers that has been used to distinguish /b/ from /d/ from /g/.

A somewhat different neural network architecture for speech recognition, which combines the concept of a matched filter with a conventional neural network to account for the dynamic within speech, is shown in Figure 2.51 [15]. The “acoustic features” of the speech are estimated via conventional neural network architectures; the pattern classifier takes the detected acoustic feature vectors (delayed appropriately) and convolves them with filters “matched” to the acoustic features and sums up the results over time. At the appropriate time (corresponding to the end of some speech unit to be detected or recognized), the output units indicate the presence of the speech.

To illustrate how the network of Figure 2.51 could be used for speech recognition, consider, as shown in Figure 2.52, a “sound” to be recognized that is characterized (in some type of sound lexicon) as the sequence of acoustic features $(\alpha, \epsilon, \delta, \beta, \gamma)$. Assume that this sound is the input to an appropriately designed network of the type shown in Figure 2.51, and the input is as shown in the first line of Figure 2.52. When the acoustic feature α is detected (as indicated by the line labeled $D_\alpha(t)$), it is delayed and then convolved with a matched filter with a long time spreading function, yielding the signal $D_\alpha(t - \tau) * P_\alpha(\tau)$ as shown in the next line of the figure. Similarly acoustic features $\epsilon, \delta, \beta,$ and γ are detected, delayed appropriately, and convolved with the appropriate matched filter, as shown in the

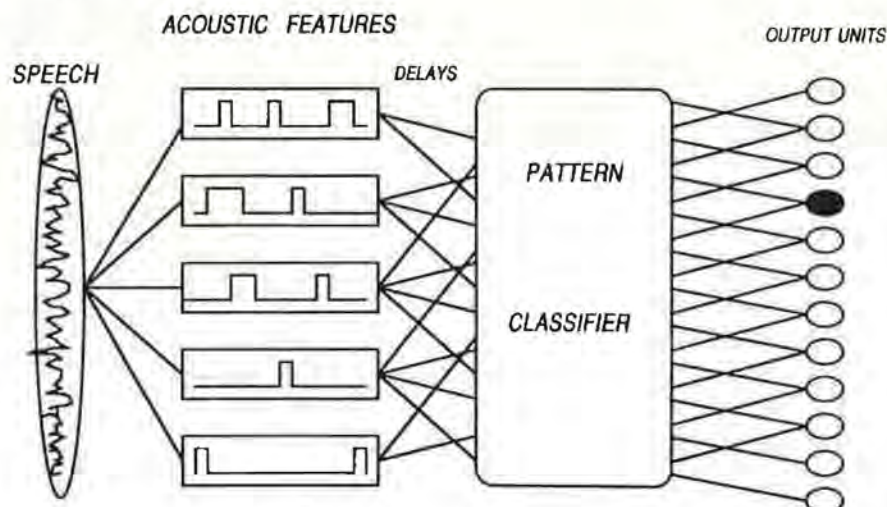


Figure 2.51 A combination neural network and matched filter for speech recognition (after Tank & Hopfield [15]).

succeeding lines in Figure 2.52. Finally, at the end of the sequence, the convolved outputs are summed up and yield a large value, indicating the recognition of the appropriate sound.

Finally, yet a third way of integrating temporal information into a neural network is shown in Figure 2.53. This network is called a hidden control neural network (HCNN) [16] and uses the time varying control, c , as a supplement to the standard input, x , to allow the network properties (input-output relations) to change over time in a well-prescribed manner.

2.6 SUMMARY

In this chapter we have presented a brief discussion of the basic speech-production/perception mechanism in human beings, and we have illustrated how we can exploit the so-called acoustic-phonetic properties of speech to identify basic sounds. Acoustic-phonetics is the broad underpinning of all speech-recognition work. Differences in approach lie in the degree of reliance on how much acoustic-phonetics can be used in the recognition process. At one extreme is the class of acoustic-phonetic recognition methods that places total reliance on the acoustic-phonetic mapping; at the other extreme is the class of pattern-recognition approaches that do not make a priori assumptions on the phonetic characteristics and instead choose to “relearn” the appropriate acoustic-phonetic mapping for specific word vocabularies and tasks via an appropriately designed training set. Finally, there is the hybrid class of artificial intelligence approaches that exploit, in various degrees, aspects of both extreme views of the speech-recognition process. We also discussed the fundamentals of neural networks, which can be considered a separate structural approach, as well as a new pattern classifier design, with potential to benefit or advance all three classical approaches described in this chapter.

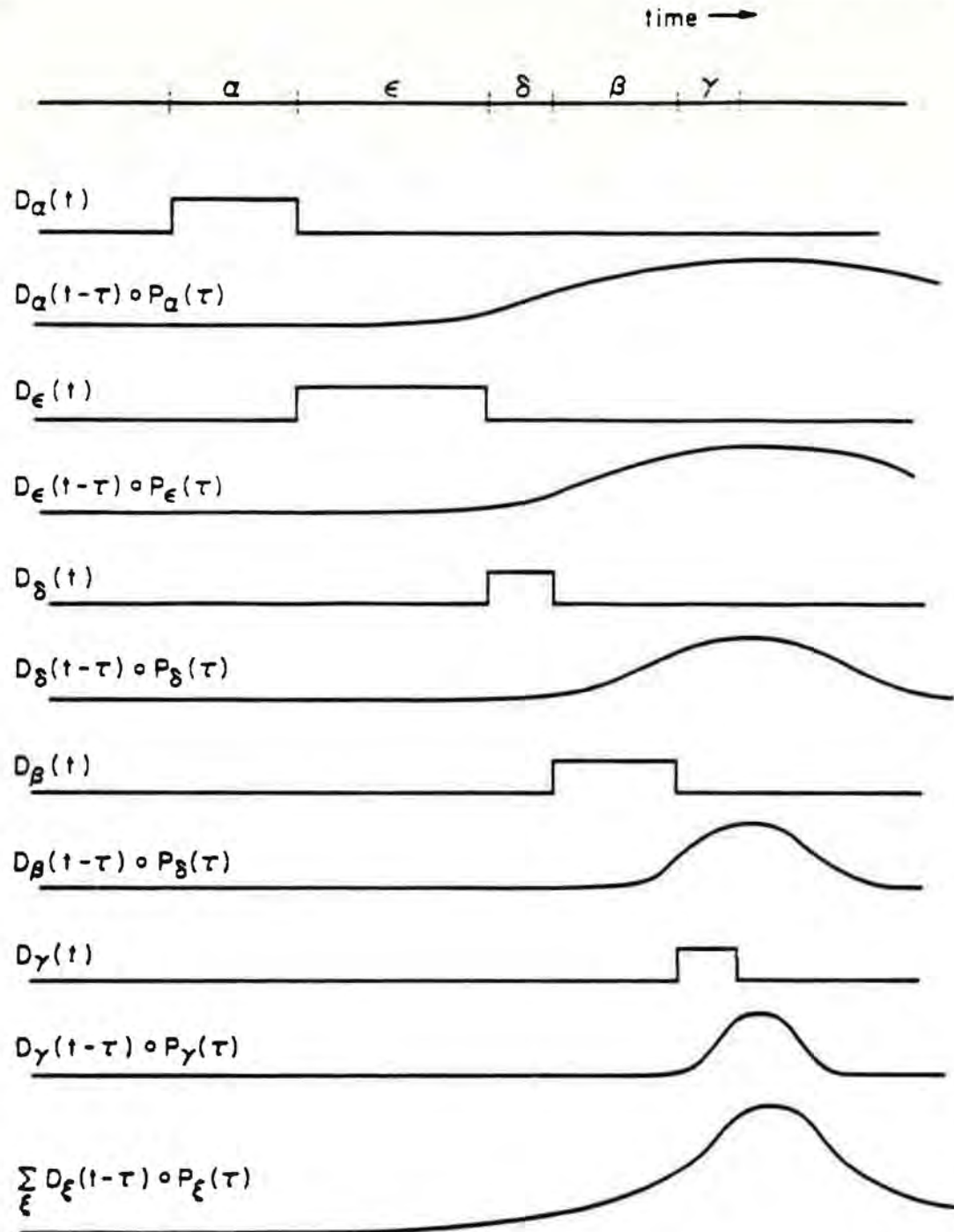


Figure 2.52 Example illustrating the combination of a neural network and a set of matched filters (after Tank & Hopfield [15]).

In the remainder of this book we will primarily discuss aspects of the pattern-recognition approach to speech recognition. However, the alternative methods will always be lurking just below the surface of our discussion.

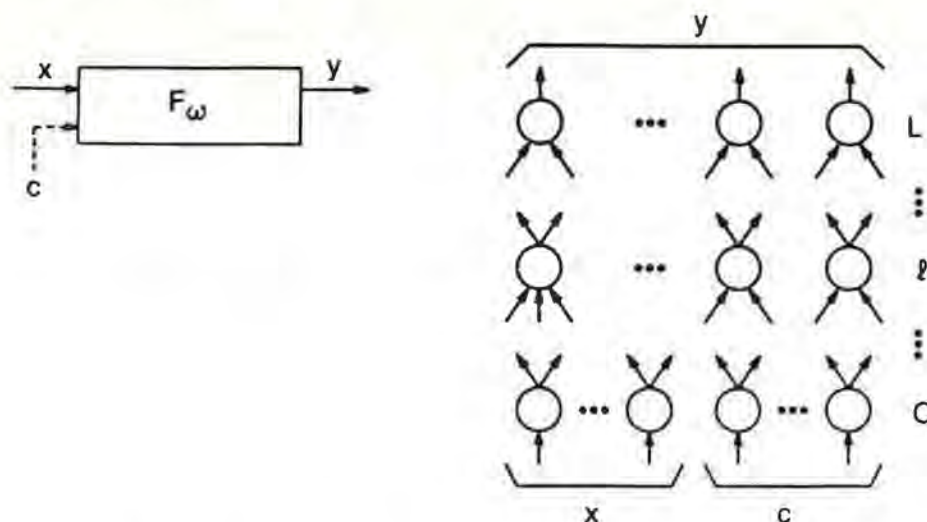


Figure 2.53 The hidden control neural network (after Levin [16]).

REFERENCES

- [1] L.R. Rabiner and S.E. Levinson, "Isolated and Connected Word Recognition—Theory and Selected Applications," *IEEE Trans. Communications*, COM-29 (5): 621–659, May 1981.
- [2] J.L. Flanagan, C.H. Coker, L.R. Rabiner, R.W. Schafer, and N. Umeda, "Synthetic Voices for Computers," *IEEE Spectrum*, 7 (10): 22–45, October 1970.
- [3] J.L. Flanagan, *Speech Analysis, Synthesis, and Perception*, 2nd ed., Springer-Verlag, New York, 1972.
- [4] K. Ishizaka and J.L. Flanagan, "Synthesis of Voiced Sounds from a Two-Mass Model of the Vocal Cords," *Bell System Tech. J.*, 50 (6): 1233–1268, July–Aug., 1972.
- [5] B.S. Atal and S.L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *J. Acoust. Soc. Am.*, 50 (2): 637–655, August 1971.
- [6] J.E. Shoup, "Phonological Aspects of Speech Recognition," 125–138, Ch. 6 in *Trends in Speech Recognition*, W. A. Lea, Ed., Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [7] G.E. Peterson and H.L. Barney, "Control Methods Used in a Study of the Vowels," *J. Acoust. Soc. Am.*, 24 (2): 175–194, March 1952.
- [8] L.R. Rabiner, "Speech Synthesis by Rule—An Acoustic Domain Approach," PhD. thesis, MIT, Cambridge, MA, June 1967.
- [9] A. Holbrook and G. Fairbanks, "Diphthong Formants and Their Movements," *J. Speech Hearing Res.*, 5 (1): 38–58, March 1962.
- [10] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth & Brooks, Monterey, CA, 1984.
- [11] V.R. Lesser, R.D. Fennell, L.D. Erman, and D.R. Reddy, "Organization of the Hearsay-II Speech Understanding System," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-23 (1): 11–23, 1975.
- [12] W.S. McCullough and W.H. Pitts, "A Logical Calculus of Ideas Immanent in Nervous

Activity," *Bull Math Biophysics*, 5: 115–133, 1943.

- [13] R. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Mag.*, 4 (2): 4–22, April 1987.
- [14] A. Weibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme Recognition Using Time Delay Neural Networks," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-37: 328–339, 1989.
- [15] D.W. Tank and J.J. Hopfield, "Neural Computation by Concentrating Information in Time," *Proc. Nat. Academy Sciences*, 84: 1896–1900, April 1987.
- [16] E. Levin, "Word Recognition Using Hidden Control Neural Architecture," *Proc. ICASSP 90*, 433–436, Albuquerque, NM, April 1990.

LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

8.1 INTRODUCTION

Throughout this book we have developed a wide range of tools, techniques, and algorithms for attacking several fundamental problems in speech recognition. In the previous chapter we saw how the different techniques came together to solve the connected word recognition problem. In this chapter we extend the concepts to include issues needed to solve the large vocabulary, continuous speech recognition problem. We will see that the fundamental ideas need modification because of the use of subword speech units; however, a great deal of the formalism for recognition, based on word units, is still preserved.

The standard approach to large vocabulary continuous speech recognition is to assume a simple probabilistic model of speech production whereby a specified word sequence, W , produces an acoustic observation sequence Y , with probability $P(W, Y)$. The goal is then to decode the word string, based on the acoustic observation sequence, so that the decoded string has the maximum a posteriori (MAP) probability, i.e.,

$$\hat{W} \ni P(\hat{W}|Y) = \max_W P(W|Y). \quad (8.1)$$

Using Bayes' Rule, Equation (8.1) can be written as

$$P(W|Y) = \frac{P(Y|W)P(W)}{P(Y)}. \quad (8.2)$$

Since $P(Y)$ is independent of W , the MAP decoding rule of Eq. (8.1) is

$$\hat{W} = \arg \max_W P(Y|W)P(W). \quad (8.3)$$

The first term in Eq. (8.3), $P(Y|W)$, is generally called the acoustic model, as it estimates the probability of a sequence of acoustic observations, conditioned on the word string. The way in which we compute $P(Y|W)$, for large vocabulary speech recognition, is to build statistical models for subword speech units, build up word models from these subword speech unit models (using a lexicon to describe the composition of words), and then postulate word sequences and evaluate the acoustic model probabilities via standard concatenation methods. Such methods are discussed in Sections 8.2–8.4 of this chapter.

The second term in Eq. (8.3), $P(W)$, is generally called the language model, as it describes the probability associated with a postulated sequence of words. Such language models can incorporate both syntactic and semantic constraints of the language and the recognition task. Often, when only syntactic constraints are used, the language model is called a grammar and may be of the form of a formal parser and syntax analyzer, an N -gram word model ($N = 2, 3, \dots$), or a word pair grammar of some type. Generally such language models are represented in a finite state network so as to be integrated into the acoustic model in a straightforward manner. We discuss language models further in Section 8.5 of this chapter.

We begin the chapter with a discussion of subword speech units. We formally define subword units and discuss their relative advantages (and disadvantages) as compared to whole-word models. We next show how we use standard statistical modeling techniques (i.e., hidden Markov models) to model subword units based on either discrete or continuous densities. We then show how such units can be trained automatically from continuous speech, without the need for a bootstrap model of each of the subword units. Next we discuss the problem of creating and implementing word lexicons (dictionaries) for use in both training and recognition phases. To evaluate the ideas discussed in this chapter we use a specified database access task, called the DARPA Resource Management (RM) task, in which there is a word vocabulary of 991 words (plus a silence or background word), and any one of several word grammars can be used. Using such a system, we show how a basic set of subword units performs on this task. Several directions for creating subword units which are more specialized are described, and several of these techniques are evaluated on the RM task. Finally we conclude the chapter with a discussion of how task semantics can be applied to further constrain the recognizer and improve overall performance.

8.2 SUBWORD SPEECH UNITS

We began Chapter 2 with a discussion of the basic phonetic units of language and discussed the acoustic properties of the phonemes in different speech contexts. We then argued that the acoustic variability of the phonemes due to context was sufficiently large and not well understood, that such units would not be useful as the basis for speech models for recognition. Instead, we have used whole-word models as the basic speech unit, both for

isolated word recognition systems and for connected word recognition systems, because whole words have the property that their acoustic representation is well defined, and the acoustic variability occurs mainly in the region of the beginning and the end of the word. Another advantage of using whole-word speech models is that it obviates the need for a word lexicon, thereby making the recognition structure inherently simple.

The disadvantages of using whole-word speech models for continuous speech recognition are twofold. First, to obtain reliable whole-word models, the number of word utterances in the training set needs to be sufficiently large, i.e., each word in the vocabulary should appear in each possible phonetic context several times in the training set. In this way the acoustic variability at the beginning and at the end of each word can be modeled appropriately. For word vocabularies like the digits, we know that each digit can be preceded and followed by every other digit; hence for an 11-digit vocabulary (zero to nine plus oh), there are exactly 121 phonetic contexts (some of which are essentially identical). Thus with a training set of several thousand digit strings, it is both realistic and practical to see every digit in every phonetic context several times. Now consider a vocabulary of 1000 words with an average of 100 phonetic contexts for both the beginning and end of each word. To see each word in each phonetic context exactly once requires $100 \times 1000 \times 100 = 10$ million carefully designed sentences. To see each combination 10 times requires 100 million such sentences. Clearly, the recording and processing of such homogeneous amounts of speech data is both impractical and unthinkable. Second, with a large vocabulary the phonetic content of the individual words will inevitably overlap. Thus storing and comparing whole-word patterns would be unduly redundant because the constituent sounds of individual words are treated independently, regardless of their identifiable similarities. Hence some more efficient speech representation is required for such large vocabulary systems. This is essentially the reason we use subword speech units.

There are several possible choices for subword units that can be used to model speech. These include the following:

- **Phonelike units (PLUs)** in which we use the basic phoneme set (or some appropriately modified set) of sounds but recognize that the acoustic properties of these units could be considerably different than the acoustic properties of the “basic” phonemes [1–7]. This is because we define the units based on linguistic similarity but model the unit based on acoustic similarity. In cases in which the acoustic and phonetic similarities are roughly the same (e.g., stressed vowels) then the phoneme and the PLU will be essentially identical. In other cases there can be large differences and a simple one-to-one correspondence may be inadequate in terms of modeling accuracy. Typically there are about 50 PLUs for English.
- **Syllable-like units** in which we again use the linguistic definition of a syllable (namely a vowel nucleus plus the optional initial and final consonants or consonant clusters) to initially define these units, and then model the unit based on acoustic similarity. In English there are approximately 10,000 syllables.
- **Dyad or demisyllable-like units** consisting of either the initial (optional) consonant cluster and some part of the vowel nucleus, or the remaining part of the vowel nucleus and the final (optional) consonant cluster [8]. For English there is something on the

order of 2000 demisyllable-like units.

- **Acoustic units**, which are defined on the basis of clustering speech segments from a segmentation of fluent, unlabeled speech using a specified objective criterion (e.g., maximum likelihood) [9]. Literally a codebook of speech units is created whose interpretation, in terms of classical linguistic units, is at best vague and at worst totally nonexistent. It has been shown that a set of 256–512 acoustic units is appropriate for modeling a wide range of speech vocabularies.

Consider the English word *segmentation*. Its representation according to each of the above subword unit sets is

- **PLUs**: /s/ /ɛ/ /g/ /m/ /ə/ /n/ /t/ /eʃ/ /ə/ /n/ (11 units)
- **syllables**: /seg/ /men/ /ta/ /tion/ (4 syllables)
- **demisyllables**: /sɛ/ /ɛg/ /mə/ /ən/ /teʃ/ /eʃ/ /ə/ /n/ (8 demisyllables)
- **acoustic units**: 17 111 37 3 241 121 99 171 37 (9 acoustic units).

We see, from the above example, that the number of subword units for this word can be as small as 4 (from a set of 10,000 units) or as large as 11 (from a set of 50 units).

Since each of the above subword unit sets is capable of representing any word in the English language, the issues in the choice of subword unit sets are the context sensitivity and the ease of training the unit from fluent speech. (In addition, for acoustic units, an issue is the creation of a word lexicon since the units themselves have no inherent linguistic interpretation.) It should be clear that there is no ideal (perfect) set of subword units. The PLU set is extremely context sensitive because each unit is potentially affected by its predecessors (one or more) and its followers. However, there is only a small number of PLUs and they are relatively easy to train. On the other extreme are the syllables which are longest units and are the least context sensitive. However, there are so many of them that they are almost as difficult to train as whole-word models.

For simplicity we will initially assume that we use PLUs as the basic speech units. In particular we use the set of 47 PLUs shown in Table 8.1 (which includes an explicit symbol for silence –h#). For each PLU we show an orthographic symbol (e.g., aa) and a word associated with the symbol (e.g., father). (These symbols are essentially identical to the ARPAPET alphabet of Table 2.1; lowercase symbols are used throughout this chapter for consistency with the DARPA community.) Table 8.2 shows typical pronunciations for several words from the DARPA RM task in terms of the PLUs in Table 8.1. A strong advantage of using PLUs is the ease of creating word lexicons of the type shown in Table 8.2 from standard (electronic) dictionaries. We will see later in this chapter how we exploit the advantages of PLUs, while reducing the context dependencies, by going to more specialized PLUs which take into consideration either the left or right (or both) contexts in which the PLU appears.

One problem with word lexicons of the type shown in Table 8.2 is that they don't easily account for variations in word pronunciation across different dialects and in the context of a sentence. Hence a simple word like "a" is often pronounced as /ey/ in isolation (e.g., the

TABLE 8.1. Set of basic PLUs for speech.

Number	Symbol	Word	Number	Symbol	Word
1	h#	silence	26	k	kick
2	aa	father	27	l	led
3	ae	bat	28	m	mom
4	ah	butt	29	n	no
5	ao	bought	30	ng	sing
6	aw	bough	31	ow	boat
7	ax	again	32	oy	boy
8	axr	diner	33	p	pop
9	ay	bite	34	r	red
10	b	bob	35	s	sis
11	ch	church	36	sh	shoe
12	d	dad	37	t	tot
13	dh	they	38	th	thief
14	eh	bet	49	uh	book
15	el	bottle	40	uw	boot
16	en	button	41	v	very
17	er	bird	42	w	wet
18	ey	bait	43	y	yet
19	f	fief	44	z	zoo
20	g	gag	45	zh	measure
21	hh	hag	46	dx	butter
22	ih	bit	47	nx	center
23	ix	roses			
24	iy	beat			
25	jh	judge			

TABLE 8.2. Typical word pronunciations (word lexicon) based on context-independent PLUs.

Word	Number of phones	Transcription				
a	1	ax				
above	4	ax	b	ah	v	
bad	3	b	ae	d		
carry	4	k	ae	r	iy	
define	5	d	iy	f	ay	n
end	3	eh	n	d		
gone	3	g	ao	n		
hours	4	aw	w	axr	z	

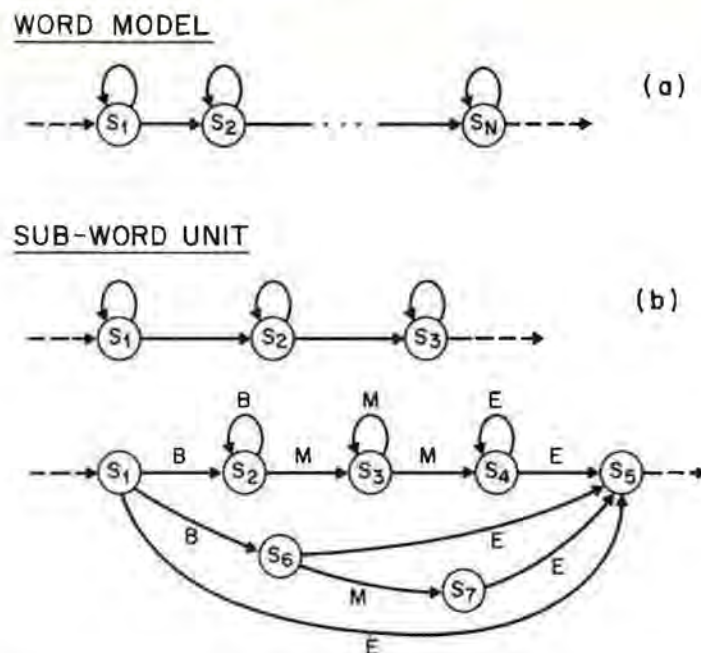


Figure 8.1 HMM representations of a word (a) and a subword unit (b).

letter A), but is pronounced as /ax/ in context. Another example is a word like “data,” which can be pronounced as /d ey t ax/ or /d ae t ax/ depending on the speaker’s dialect. Finally words like “you” are normally pronounced as /y uw/ but in context often are pronounced as /jh ax/ or /jh uh/. There are several ways of accounting for word pronunciation variability, including multiple entries in the word lexicon, use of phonological rules in the recognition grammar, and use of context dependent PLUs. We will discuss these options later in this chapter.

8.3 SUBWORD UNIT MODELS BASED ON HMMs

As we have shown several times in this book, the most popular way in which speech is modeled is as a left-to-right hidden Markov model. As shown in Figure 8.1a, a whole-word model typically uses a left-to-right HMM with N states, where N can be a fixed value (e.g., 5–10 for each word), or can be variable with the number of sounds (phonemes) in the word, or can be set equal to the average number of frames in the word. For subword units, typically, the number of states in the HMM is set to a fixed value, as shown in Figure 8.1b where a three-state model is used. This means that the shortest tokens of each subword unit must last at least three frames, a restriction that seems reasonable in practice. (Models that use jumps to eliminate this restriction have been studied [2].)

To represent the spectral density associated with the states of each subword unit, one of three approaches can be used. These approaches are illustrated in Figure 8.2. Perhaps the simplest approach is to design a VQ-based codebook for all speech sounds (as shown in part a of the figure). For this approach the probability density of the observed

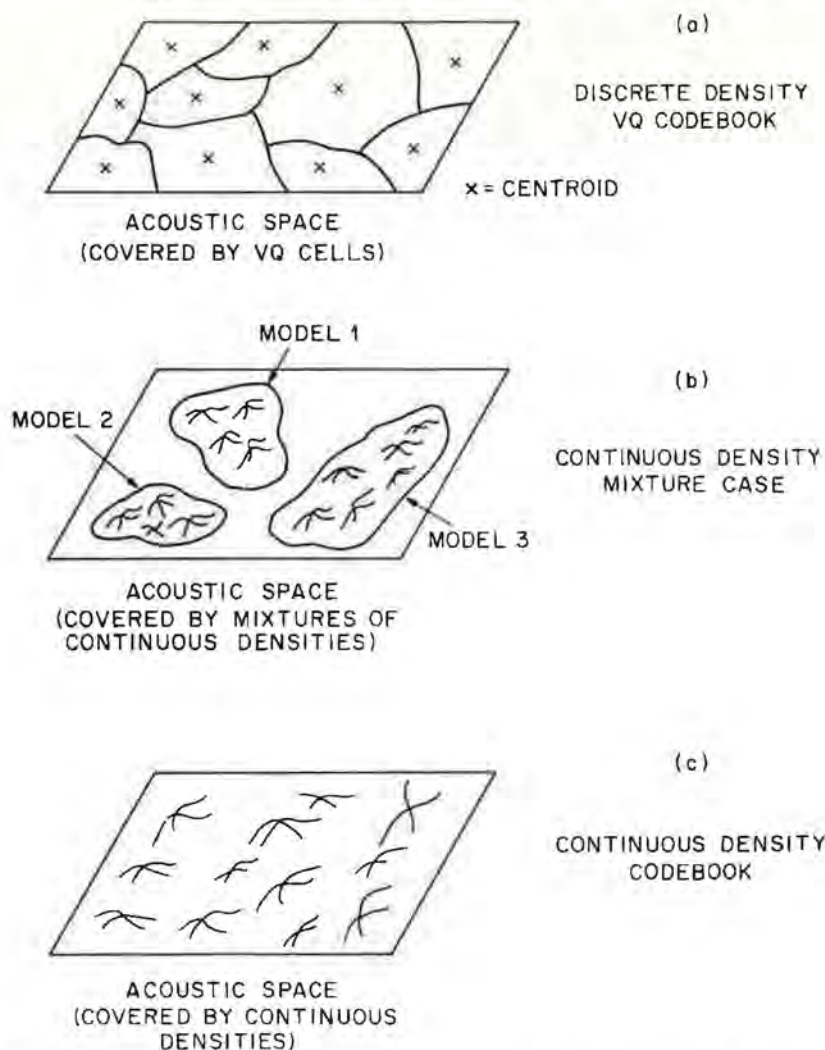


Figure 8.2 Representations of the acoustic space of speech by (a) partitioned VQ cells, (b) sets of continuous mixture Gaussian densities, and (c) a continuous-density codebook (after Lee et al. [7]).

spectral sequence within each state of each PLU is simply a discrete density defined over the codebook vectors. The interpretation of the discrete density within a state is that of implicitly isolating the part of the acoustic space in which the spectral vectors occur and assigning the appropriate codebook vector (over that part of the space) a fixed probability for spectral vectors within each isolated region regardless of its proximity to the corresponding codebook vector. A second alternative, illustrated in part b of Figure 8.2, is to represent the continuous probability density in each subword unit state by a mixture density that explicitly defines the part of the acoustic space in which the spectral vectors occur. Each mixture component has a spectral mean and variance that is highly dependent on the spectral characteristics of the subword unit (i.e., highly localized in the acoustic space). Hence the models for different subword units usually do not have substantial overlap in the acoustic space. Finally, a third alternative is to design a type of continuous density codebook over the entire acoustic space, as illustrated in part c of Figure 8.2. Basically the entire acoustic

space is covered by a set of independent Gaussian densities, derived in much the same way as the discrete VQ codebook, with the resulting set of means and covariances stored in a codebook. This alternative is a compromise between the previous two possibilities. It differs from the discrete density case in the way the probability of an observation vector is computed; instead of assigning a fixed probability to any observation vector that falls within an isolated region, it actually determines the probability according to the closeness of the observation vector to the codebook vector (i.e., it calculates the exponents of the Gaussian distributions). For each state of each subword unit, the density is assumed to be a mixture of the fixed codebook densities. Hence, even though each state is characterized by a continuous mixture density, one need only estimate the set of mixture gains to specify the continuous density completely. Furthermore, since the codebook set of Gaussian densities is common for all states of all subword models, one can precompute the likelihoods associated with an input spectral vector for each of the codebook vectors, and ultimately determine state likelihoods using only a simple dot product with the state mixture gains. This represents a significant computational reduction over the full mixture continuous density case. This mixed density method has been called the tied mixture approach [10, 28] as well as the semicontinuous modeling method [11] and has been applied to the entire acoustic space as well as to pieces of the acoustic space for detailed PLU modeling. This method can be further extended to the case in which a set of continuous density codebooks is designed, one for each state of each basic (context independent) speech unit. One can then estimate sets of mixture gains appropriate to context dependent versions of each basic speech unit and use them appropriately for recognition. We will return to this issue later in this chapter.

8.4 TRAINING OF SUBWORD UNITS

Implicitly it would seem that training of the models for subword units would be extremely difficult, because there is no simple way to create a bootstrap model of such short, imprecisely defined, speech sounds. Fortunately, this is not the case. The reason for this is because of the inherent tying of subword units across words and sentences—that is, every subword unit occurs a large number of times in any reasonable size training set. Hence estimation algorithms like the forward-backward procedure, or the segmental k -means algorithm, can start with a uniform segmentation (flat or random initial models) and rapidly converge to the best model estimates in just a few iterations.

To illustrate how models of subword units are estimated, assume we have a labeled training set of speech sentences, where each sentence consists of the speech waveform and its transcription into words. (We assume that waveform segmentation into words is not available.) We further assume a word lexicon is available that provides a transcription of every word in the training set strings in terms of the set of subword units being trained. We assume that silence can (but needn't) precede or follow any word within a sentence (i.e., we allow pauses in speaking), with silence at the beginning and end of each sentence the most likely situation. Based on the above assumptions, a typical sentence in the training set can be transcribed as

$$S_W : W_1 W_2 W_3 \dots W_{I_1}$$

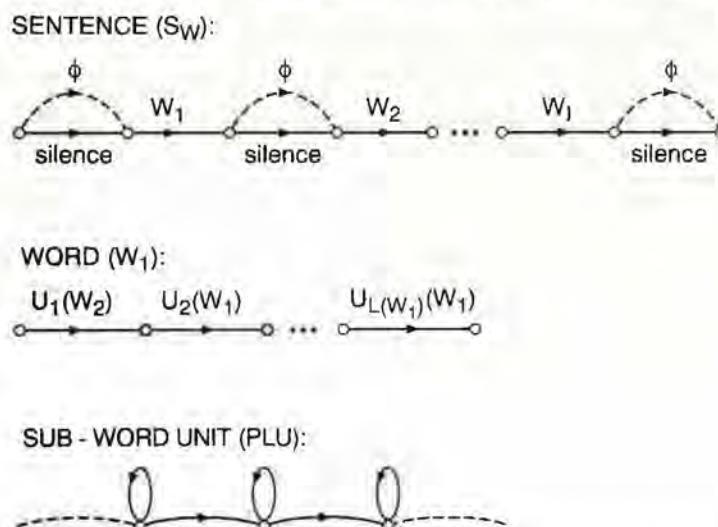


Figure 8.3 Representation of a sentence, word, and subword unit in terms of FSNs.

in which each W_i , $1 \leq i \leq I$, is a word in the lexicon. Hence the sentence “show all alerts” is a three-word sentence with $W_1 = \text{show}$, $W_2 = \text{all}$, and $W_3 = \text{alerts}$. Each word can be looked up in the lexicon to find its transcription in terms of subword units. Hence the sentence S can be written in terms of subword units as

$$S_U : U_1(W_1)U_2(W_1) \dots U_{L(W_1)}(W_1) \oplus U_1(W_2)U_2(W_2) \dots U_{L(W_2)}(W_2) \oplus \\ U_1(W_3)U_2(W_3) \dots U_{L(W_3)}(W_3) \oplus \dots \oplus U_1(W_I)U_2(W_I) \dots U_{L(W_I)}(W_I),$$

where $L(W_1)$ is the length (in units) of word W_1 , etc. Finally we replace each subword unit by its HMM (the three-state models shown in Figure 8.1) and incorporate the assumptions about silence between words to give an extended HMM for each sentence.

The above process is illustrated (in general) in Figure 8.3. We see that a sentence is represented as a finite-state network (FSN) where the arcs are either words or silence or null arcs (where a null (ϕ) transition is required to skip the alternative silence). Each word is represented as an FSN of subword units and each subword unit is represented as a three-state HMM.

Figure 8.4 shows the process of creating the composite FSN for the sentence “Show all alerts,” based on a single-word pronunciation lexicon. One feature of this implementation is the use of a single-state HMM for the silence word. This is used (rather than the three-state HMMs used for each PLU), since silence is generally stationary and has no temporal structure to exploit.

When there are multiple representations of words in the lexicon (e.g., for two or more distinct pronunciations) it is easy to modify the FSN of Figures 8.3 and 8.4 to add parallel paths for the word arcs. (We will see that only one path is chosen in training, namely the best representation of the actual word pronunciation in the context of the spoken sentence.) Furthermore, multiple models of each subword unit can be used by introducing parallel paths in the word FSNs and then choosing the best version of each subword unit in the decoding process.

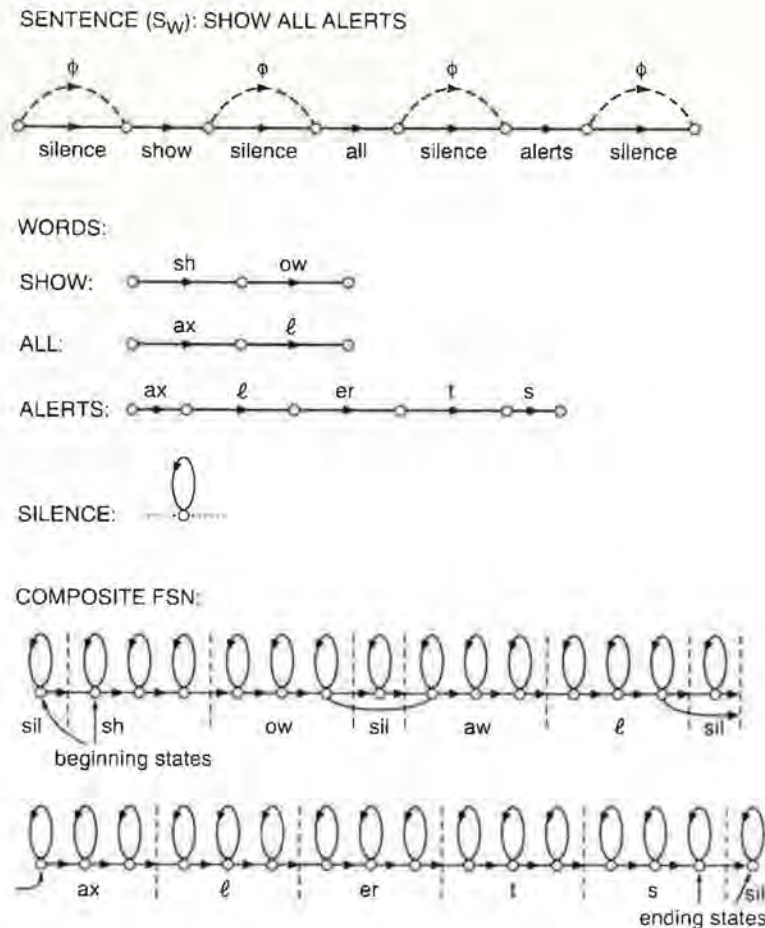


Figure 8.4 Creation of composite FSN for sentence "Show all alerts."

Once a composite sentence FSN is created for each sentence in the training set, the training problem becomes one of estimating the subword unit model parameters which maximize the likelihood of the models for all the given training data. The maximum likelihood parameters can be solved for using either the forward-backward procedure (see Ref. [2] for example) or the segmental k -means training algorithm. The way in which we use the segmental k -means training procedure to estimate the set of model parameters (based on using a mixture density with M mixtures/state) is as follows:

1. **Initialization:** Linearly segment each training utterance into units and HMM states assuming no silence between words (i.e., silence only at the beginning and end of each sentence), a single lexical pronunciation of each word, and a single model for each subword unit. Figure 8.5, iteration 0, illustrates this step for the first few units of one training sentence. Literally we assume every unit is of equal duration initially.
2. **Clustering:** All feature vectors from all segments corresponding to a given state (i) of a given subword unit are partitioned into M clusters using the k -means algorithm. (This step is iterated for all states of all subword units.)
3. **Estimation:** The mean vectors, μ_{ik} , the (diagonal) covariance matrices, U_{ik} , and the

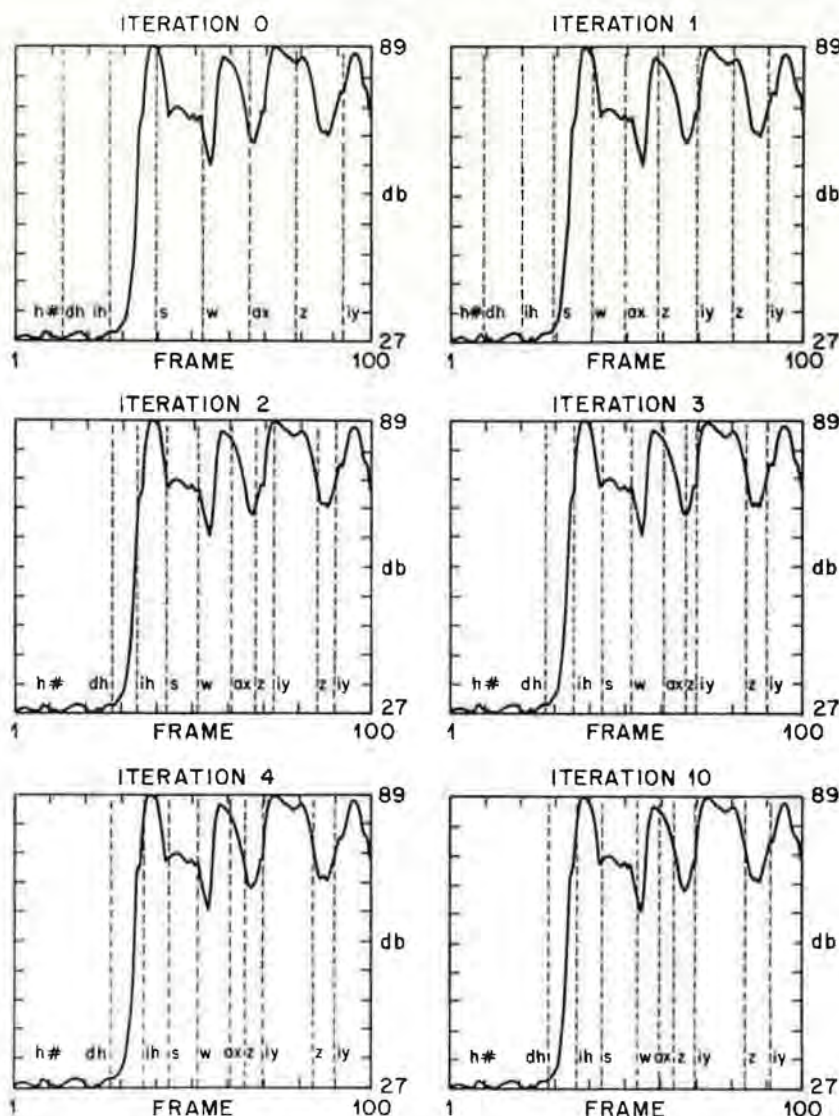


Figure 8.5 Segmentations of a training utterance resulting from the segmental k -means training for the first several iterations (after Lee et al. [7]).

mixture weights, c_{ik} , are estimated for each cluster k in state i . (This step is iterated for all states of all subword units.)

4. **Segmentation:** The updated set of subword unit models (based on the estimation of step 3) is used to resegment each training utterance into units and states (via Viterbi decoding). At this point multiple lexical entries can be used for any word in the vocabulary. Figure 8.5 shows the result of this resegmentation step for iterations 1–4 and 10 for one training utterance. It can be seen that by iteration 2 the segmentation into subword units is remarkably stable.
5. **Iteration:** Steps 2–4 are iterated until convergence (i.e., until the overall likelihoods stop increasing).

Figure 8.6 illustrates the resulting segmentation of the first few units of the utterance

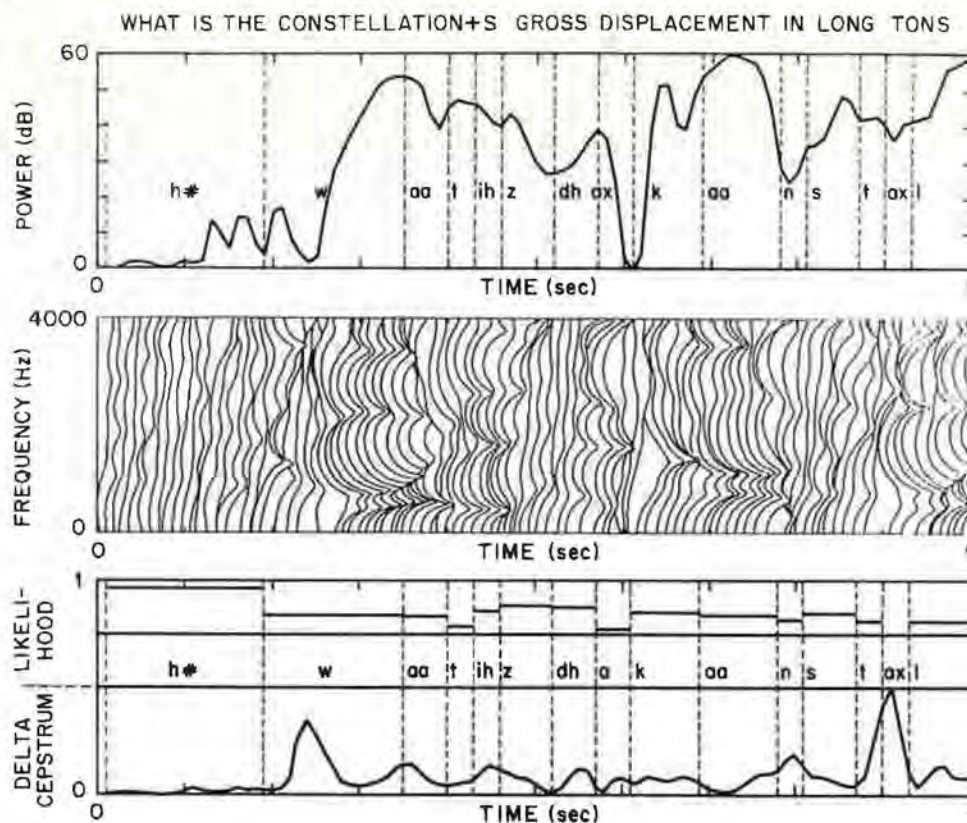


Figure 8.6 Segmentation of an utterance into PLUs (after Lee et al. [7]).

“What is the constellation. . .” Shown in this figure are the power contour in dB (upper panel), the running LPC spectral slices (the middle panel), and the likelihood scores and delta-cepstral values (lower panel) for the first second of the sentence. The resulting segmentations are generally remarkably consistent with those one might manually choose based on acoustic-phonetic criteria. Since we use an acoustic criterion for choice of segmentation points, the closeness of PLU units to true phonetic units is often remarkable, especially in light of the phonetic variability in word pronunciation discussed previously.

In summary we have shown how one can use a training set of speech sentences that have only word transcriptions associated with each sentence and optimally determine the parameters of a set of subword unit HMMs. The resulting parameter estimates are extremely robust to the training material as well as to details of word pronunciation as obtained from the word lexicon. The reason for this is that a common word lexicon (with associated word pronunciation errors) is used for both training and recognition; hence errors in associating proper subword units to words are consistent throughout the process and are less harmful than they would be in alternative methods of estimating parameters of subword models.

The results of applying the segmental k -means training procedure to a set of 3990 training sentences from 109 different talkers, in terms of PLU counts and PLU likelihood scores are shown in Table 8.3. A total of 155,000 PLUs occurred in the 3990 sentences with silence (h#) having the most occurrences (10,638 or 6.9% of the total) and nx (flapped

TABLE 8.3. PLU statistics on count and average likelihood score.

PLU	Count	%	Average likelihood	(Rank)
h#	10638	6.9	18.5	(1)
r	8997	5.8	8.4	(45)
t	8777	5.7	9.7	(37)
ax	8715	5.6	7.1	(47)
s	8625	5.6	15.4	(3)
n	8478	5.5	8.3	(46)
ih	6542	4.2	9.9	(35)
iy	5816	3.7	12.0	(17)
d	5391	3.5	8.5	(44)
ae	4873	3.1	13.3	(10)
ℓ	4857	3.1	8.9	(41)
z	4733	3.0	12.4	(14)
ch	4604	3.0	11.2	(21)
k	4286	2.8	10.6	(27)
p	3793	2.4	14.3	(6)
m	3625	2.3	8.5	(43)
ao	3489	2.2	10.4	(32)
f	3276	2.1	17.7	(2)
ey	3271	2.1	14.5	(5)
w	3188	2.1	10.2	(34)
ix	3079	2.0	8.7	(42)
dh	2984	1.9	11.8	(18)
v	2979	1.9	12.0	(16)
aa	2738	1.8	10.3	(33)
b	2138	1.4	10.7	(25)
y	2137	1.4	13.1	(11)
uw	2032	1.3	10.6	(26)
sh	1875	1.2	13.1	(12)
ow	1875	1.2	10.9	(24)
axr	1825	1.2	9.5	(38)
ah	1566	1.0	11.3	(20)
dx	1548	1.0	10.4	(31)
ay	1527	1.0	13.9	(8)
en	1478	0.9	9.1	(40)
g	1416	0.9	9.8	(36)
hh	1276	0.8	11.4	(19)
th	924	0.6	14.1	(7)
ng	903	0.6	9.1	(39)
ch	885	0.6	12.5	(13)
el	863	0.6	11.0	(23)
er	852	0.5	10.6	(29)
jh	816	0.5	10.6	(28)
aw	682	0.4	13.6	(9)
uh	242	0.2	11.0	(22)
zh	198	0.1	12.2	(15)
oy	130	0.1	15.3	(4)
nx	57	0.04	10.4	(30)

n) having the fewest occurrences (5 or 0.04% of the total). In terms of average likelihood scores, silence (h#) had the highest score (18.5) followed by f (17.7) and s (15.4), while ax had the lowest score (7.1), followed by n (8.3) and r (8.4). (Note that, in this case, a higher average likelihood implies less variation among different renditions of the particular sound.) It is interesting to note that the PLUs with the three lowest average likelihood scores (ax, n, and r) were among the most frequently occurring sounds (r was second, n sixth, and ax fourth in frequency of occurrence). Similarly, some of the sounds with the highest likelihood scores were among the least occurring sounds (e.g., oy was fourth according to likelihood score but 21st according to frequency of occurrence).

8.5 LANGUAGE MODELS FOR LARGE VOCABULARY SPEECH RECOGNITION

Small vocabulary speech-recognition systems are used primarily for command-and-control applications where the vocabulary words are essentially acoustic control signals that the system has to respond to. (See Chapter 9 for a discussion of command-and-control applications of speech recognition.) As such, these systems generally do not rely heavily on language models to accomplish their selected tasks. A large vocabulary speech-recognition system, however, is generally critically dependent on linguistic knowledge embedded in the input speech. Therefore, for large vocabulary speech recognition, incorporation of knowledge of the language, in the form of a “language” model, is essential. In this section we discuss a statistically motivated framework for language modeling.

The goal of the (statistical) language model is to provide an estimate of the probability of a word sequence W for the given recognition task. If we assume that W is a specified sequence of words, i.e.,

$$W = w_1 w_2 \dots w_Q, \quad (8.4)$$

then it would seem reasonable that $P(W)$ can be computed as

$$P(W) = P(w_1 w_2 \dots w_Q) = P(w_1)P(w_2|w_1)P(w_3|w_1 w_2) \dots P(w_Q|w_1 w_2 \dots w_{Q-1}). \quad (8.5)$$

Unfortunately, it is essentially impossible to reliably estimate the conditional word probabilities, $P(w_j|w_1 \dots w_{j-1})$ for all words and all sequence lengths in a given language. Hence, in practice, it is convenient to use an N -gram word model, where we approximate the term $P(w_j|w_1 \dots w_{j-1})$ as

$$P(w_j|w_1 w_2 \dots w_{j-1}) \approx P(w_j|w_{j-N+1} \dots w_{j-1}), \quad (8.6)$$

i.e., based only on the preceding $N - 1$ words. Even N -gram probabilities are difficult to estimate reliably for all but $N = 2$ or possibly 3. Hence, in practice, it is often convenient to use a word pair model that specifies which word pairs are valid in the language through the use of a binary indicator function, i.e.,

$$P(w_j|w_k) = \begin{cases} 1 & \text{if } w_k w_j \text{ is valid} \\ 0 & \text{otherwise} \end{cases} \quad (8.7)$$

Another simple language model, often called the no-grammar model, assumes $P(w_j|w_k) = 1$ for all j and k , so that every word is assumed capable of being followed by every other word in the language. In the next section we show how the word pair and the no-grammar models can be implemented as finite state networks so as to be integrated simply into a recognition decoding algorithm.

Alternative language models include formal grammars (e.g., context free or context dependent grammar), N -grams of word classes (rather than words) etc. These types of grammars provide more realistic models for natural language input to machines than the artificial N -grams or words, or the word pair grammars. However, they are somewhat more difficult to integrate with the acoustic decoding and hence will not be discussed here.

8.6 STATISTICAL LANGUAGE MODELING

In large vocabulary speech recognition, in which word sequences W are uttered to convey some message, the language model $P(W)$ is of critical importance to the recognition accuracy as shown in Eq. (8.3). In most cases, the language model $P(W)$ has to be estimated from a given (large) text corpus. In this section we discuss how to construct such a statistical language model from a (textual) training corpus.

For practical reasons, the word sequence probability $P(W)$ is approximated by

$$P_N(W) = \prod_{i=1}^Q P(w_i|w_{i-1}, w_{i-2}, \dots, w_{i-N+1}), \quad (8.8)$$

which is called an N -gram language model. The conditional probabilities $P(w_i|w_{i-1}, \dots, w_{i-N+1})$ can be estimated by the simple relative frequency approach,

$$\hat{P}(w_i|w_{i-1}, \dots, w_{i-N+1}) = \frac{F(w_i, w_{i-1}, \dots, w_{i-N+1})}{F(w_{i-1}, \dots, w_{i-N+1})}, \quad (8.9)$$

in which F is the number of occurrences of the string in its argument in the given training corpus. Obviously, in order for the estimate in Eq. (8.9) to be reliable, $F(w_i, w_{i-1}, \dots, w_{i-N+1})$ has to be substantial in the given corpus. The implications of this are that the size of the training corpus may be prohibitively large and that $F(w_i, w_{i-1}, \dots, w_{i-N+1}) = 0$ for many possible word strings, $w_i, w_{i-1}, \dots, w_{i-N+1}$, due to the limited size of the corpus.

One way to circumvent this problem is to smooth the N -gram frequencies as suggested by Jelinek et al. [12]. Consider $N = 3$, the trigram model. The smoothing is done by interpolating trigram, bigram and unigram relative frequencies

$$\hat{P}(w_3|w_1, w_2) = p_1 \frac{F(w_1, w_2, w_3)}{F(w_1, w_2)} + p_2 \frac{F(w_1, w_2)}{F(w_1)} + p_3 \frac{F(w_1)}{\sum F(w_i)}, \quad (8.10)$$

in which the nonnegative weights satisfy $p_1 + p_2 + p_3 = 1$ and $\sum F(w_i)$ is the size of the corpus. The weights depend on the values of $F(w_1, w_2)$ and $F(w_1)$ and can be obtained by applying the principle of cross-validation [12].

8.7 PERPLEXITY OF THE LANGUAGE MODEL

Having constructed a language model from a training corpus, one may ask how well the language model will perform in the context of speech recognition. This can be answered based on the concept of source of information in information theory. To provide such a measure of performance, we must first discuss several concepts, including entropy, estimated entropy, and perplexity.

Consider an information source that puts out sequences of words (symbols) w_1, w_2, \dots, w_Q , each of which is chosen from a vocabulary \bar{V} with size $|\bar{V}|$, according to some stochastic law. The entropy of the source can be defined as

$$H = - \lim_{Q \rightarrow \infty} \left(\frac{1}{Q} \right) \left\{ \sum P(w_1, w_2, \dots, w_Q) \log P(w_1, w_2, \dots, w_Q) \right\}, \quad (8.11)$$

in which $P(\cdot)$ is the probability of the argument string the source will put out according to the stochastic law and the summation is over all string sequences w_1, w_2, \dots, w_Q . If the words in the string sequence are generated by the source in an independent manner

$$P(w_1, w_2, \dots, w_Q) = P(w_1)P(w_2) \dots P(w_Q), \quad (8.12)$$

then

$$H = - \sum_{w \in \bar{V}} P(w) \log P(w), \quad (8.13)$$

which is sometimes referred to as the first-order entropy of the source (even if Eq. (8.12) is not true).

The quantity H of Eq. (8.11) can be considered as the average information of the source when it puts out a word w . Equivalently, a source of entropy H is one that has as much information content as a source which puts out words equiprobably from a vocabulary of size 2^H .

If the source is ergodic (meaning its statistical properties can be completely characterized in a sufficiently long sequence that the source puts out), the entropy of Eq. (8.11) is equivalent to

$$H = - \lim_{Q \rightarrow \infty} \left(\frac{1}{Q} \right) \log P(w_1, w_2, \dots, w_Q). \quad (8.14)$$

In other words, we can compute the entropy from a “typical” (long) sequence of words generated by the source. The length of this typical sequence (i.e., the corpus) has to approach infinity, which is of course unattainable. We often compute H based on a finite but sufficiently large Q ; i.e.,

$$H = - \left(\frac{1}{Q} \right) \log P(w_1, w_2, \dots, w_Q). \quad (8.15)$$

An interesting interpretation of H from the perspective of speech recognition is that it is the degree of difficulty that the recognizer encounters, on average, when it is to determine a word from the same source. This difficulty, or uncertainty, is based on the actual probability $P(w_1, w_2, \dots, w_Q)$ which, for natural languages, is usually not known

beforehand and thus has to be estimated. (We do not include acoustic uncertainty in the present context of language modeling.)

One way to estimate H is to use $P(W) = P(w_1, w_2, \dots, w_Q)$ from the language model. For example, if the N -gram language model $P_N(W)$ (Eq. (8.8)) is used, an estimate of H of Eq. (8.15) is thus

$$H_p = -\frac{1}{Q} \sum_{i=1}^Q \log P(w_i | w_{i-1}, w_{i-2}, \dots, w_{i-N+1}). \quad (8.16)$$

In general,

$$H_p = -\frac{1}{Q} \log \hat{P}(w_1, w_2, \dots, w_Q), \quad (8.17)$$

where $\hat{P}(w_1, w_2, \dots, w_Q)$ is an estimate of $P(w_1, w_2, \dots, w_Q)$. The quantity H_p is an estimated entropy as calculated from a sufficiently long sequence based on a language model. If the source is ergodic and $Q \rightarrow \infty$, $H_p \geq H$. Intuitively, this can be easily verified by the fact that knowledge of the true probability $P(w_1, w_2, \dots, w_Q)$ is the best a recognizer can use and any other probability estimate or language model can never make the task easier for the recognizer. Since H_p is an indication of the recognition difficulty lower-bounded by H , a language model that achieves a lower H_p (i.e., closer to H) is therefore considered a better model than another language model which leads to a higher H_p .

Associated with H_p is a quantity called perplexity (often called the average word branching factor of the language model) defined as

$$B = 2^{H_p} = \hat{P}(w_1, w_2, \dots, w_Q)^{-1/Q}. \quad (8.18)$$

Note that H_p is the average difficulty or uncertainty in each word based on the language model. When the recognizer uses this language model for the task, the difficulty it faces is equivalent to that of recognizing a text generated by a source that chooses words from a vocabulary size of B independently of each other and with equal probability. Another way to view perplexity is to consider it as the average number of possible words following any string of $(N - 1)$ words in a large corpus based on an N -gram language model. Perplexity is an important parameter in specifying the degree of sophistication in a recognition task, from the source uncertainty to the quality of the language model.

8.8 OVERALL RECOGNITION SYSTEM BASED ON SUBWORD UNITS

A block diagram of the overall continuous speech-recognition system based on subword speech units is shown in Figure 8.7. The first step in the processing is spectral analysis to derive the feature vector used to characterize the spectral properties of the speech input. For the most part, we will consider spectral vectors with 38 components consisting of 12 cepstral components, 12 delta cepstral components, 12 delta-delta cepstral components, delta log energy, and delta-delta log energy. (Systems with the first 12 and the first 24

CONTINUOUS SENTENCE RECOGNIZER

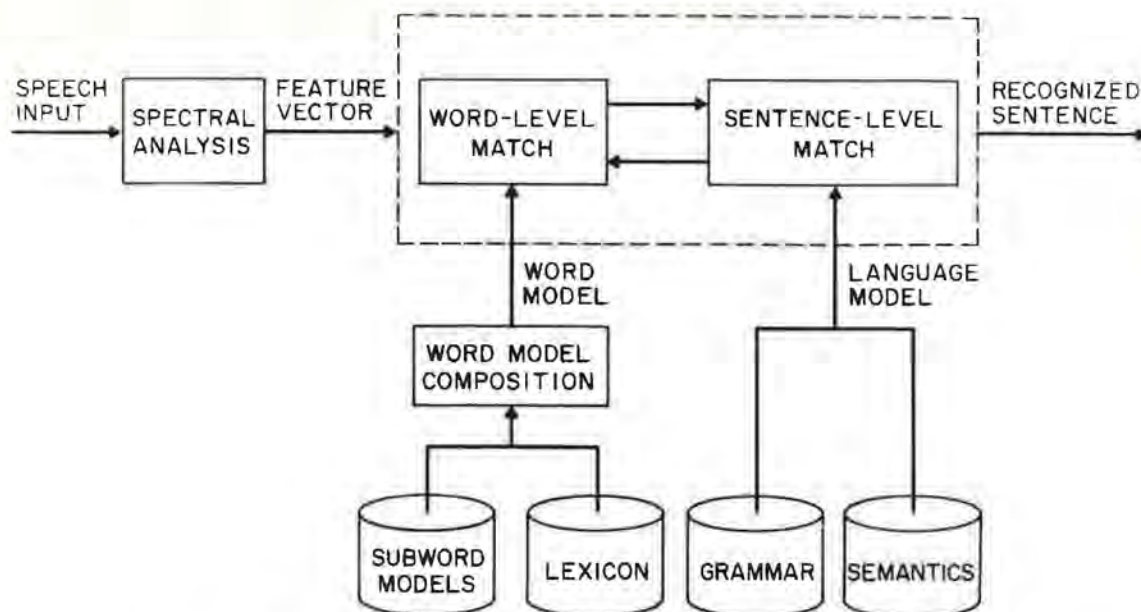


Figure 8.7 Overall block diagram of subword unit based continuous speech recognizer.

features were also studied, but results on such systems will not be presented here.)

The second step in the recognizer is a combined word-level/sentence-level match. The way this is accomplished is as follows. Using the set of subword HMMs and the word lexicon, a set of word models (HMMs) is created by concatenating each of the subword unit HMMs as specified in the word lexicon. At this point, the system is very similar to the connected word recognizers of Chapter 7. The way in which the sentence-level match is done is via an FSN realization of the word grammar (the syntax of the system) and the semantics as expressed in a composite FSN language model. The implementation of the combined word-level match/sentence-level match is via any of the structures described in Chapter 7. In particular, most systems use structures similar to the frame synchronous level-building method (usually with some type of beam search to restrict the range of paths) to solve for the “best” recognition sentence.

Consider using the recognizer of Figure 8.7 for a database management task called the Naval Resource (Battleship) Management Task—as popularly defined within the DARPA community [13]. This task, which has a 991-word vocabulary (plus a separate silence word), can be used to query a database as to locations, attributes, constraints, history, and other information about ships within the database. Typical examples of sentences used to query the database include

- what is mishawaka’s percent fuel
- total the ships that will arrive in diego-garcia by next month

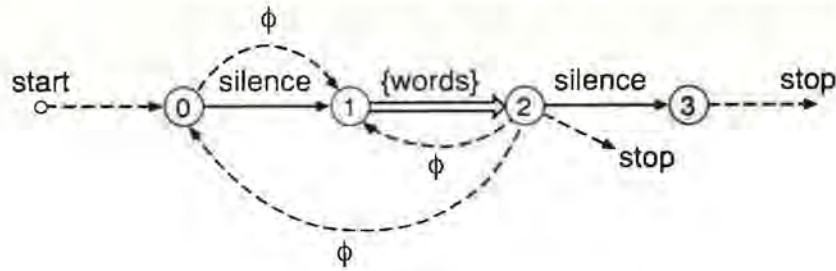


Figure 8.8 FSN for the NG syntax.

- do any vessels that are in gulf of tonkin have asw mission area of m4
- show the names of any submarines in yellow sea on twenty eight october
- list all the alerts
- what's jason's m-rating on mob
- give t-lam vessels that weren't deployed in november.

The vocabulary thus includes many jargon words, such as m4, m-rating, mob, and t-lam, and several long-content words, such as mishawaka's, diego-garcia, submarines, november, etc., and many short-function words, such as is, the, by, do, in, of, and on.

A wide range of sentences can be constructed from the 991-word vocabulary to query this database. It is possible to construct a finite-state network representation of the full grammar associated with all such sentences. The perplexity (average word branching factor) (see Section 8.7) of the full grammar network is computed to be about 9. However, such a network is rather large (because of the high degree of constraint among words within the vocabulary which form syntactically valid and semantically meaningful sentences) with upward of 50,000 arcs and 20,000 nodes, and cannot easily be implemented as a practical system. Instead, several types of FSN approximations to the full grammar have been constructed.

Perhaps the least constraining grammar (and the simplest to implement) is the no grammar (NG) case, in which any word in the vocabulary is allowed to follow any word in the vocabulary. Such an FSN has the property that, although its coverage of valid sentences is perfect, its overcoverage of the language (i.e., the ratio of sentences generated by the grammar to valid sentences within the task language) is extremely large. The perplexity of the FSN for the NG case is 991, since each word can follow every word in the grammar (assuming all words are essentially equiprobable). The FSN for the NG case is shown in Figure 8.8. (Note that the FSN of Figure 8.8 allows arbitrary phrasing, i.e., groups of words spoken together followed by a pause, because of the silence model and the null arcs.)

A second FSN form of the task syntax is to create a word pair (WP) grammar that specifies explicitly which words can follow each of the 991 words in the vocabulary. The perplexity of this grammar is about 60, and the overcoverage, while significantly below that of the NG case, is still very high. Although the network of Figure 8.8 could be used for the WP grammar (by explicitly including the word pair information at node 2), a somewhat more efficient structure exploits the fact that only a subset of the vocabulary occurs as the first word in a sentence (*B* or beginning words), and only a subset of the vocabulary occurs

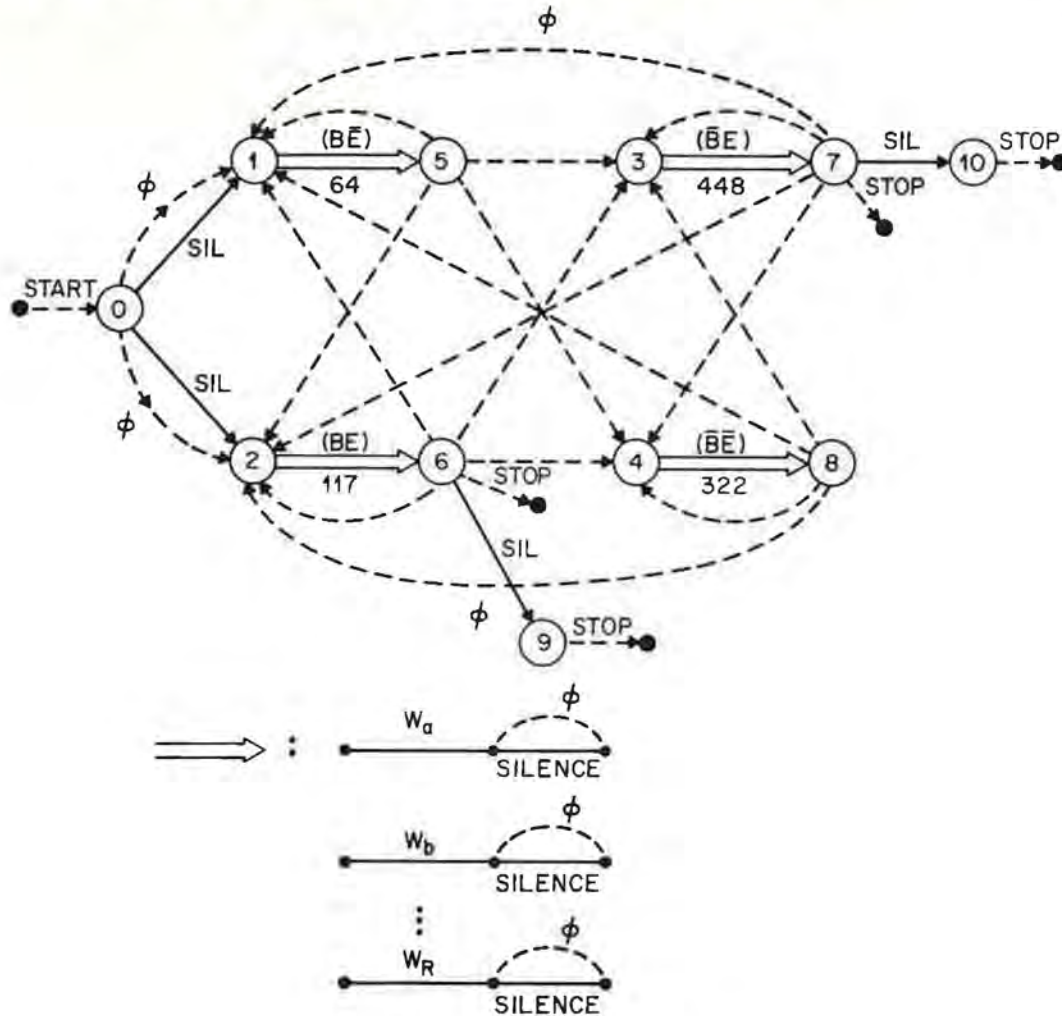


Figure 8.9 FSN of the WP syntax.

as the last word in a sentence (E or ending words); hence we can partition the vocabulary into four nonoverlapping sets of words, namely

- $\{BE\}$ =set of words that can either begin or end a sentence, $|BE| = 117$
- $\{B\bar{E}\}$ =set of words that can begin a sentence but cannot end a sentence, $|B\bar{E}| = 64$
- $\{\bar{B}E\}$ =set of words that cannot begin a sentence but can end a sentence, $|\bar{B}E| = 488$
- $\{\bar{B}\bar{E}\}$ =set of words that cannot begin or end a sentence, $|\bar{B}\bar{E}| = 322$.

The resulting FSN, based on this partitioning scheme, is shown in Figure 8.9. This network has 995 real arcs and 18 null arcs. To account for silence between words (which is optional), each word arc bundle (e.g., nodes 1 to 4) is expanded to individual words followed by optional silence, as shown at the bottom of Figure 8.9. Hence the overall FSN allows recognition of sentences of the form

$$S : (\text{silence}) - \{B\bar{E}, BE\} - (\text{silence}) - (\{W\}) \dots (\{W\}) - (\text{silence}) - \{\bar{B}E, BE\} - (\text{silence}).$$

Finally, one could construct a task syntax based on statistical word bigram (or even

trigram) probabilities—that is, we assign a probability, p_{ij} , to each word pair (W_i, W_j) where p_{ij} is the probability that W_i is followed immediately by W_j . That is, if W_n is the n^{th} word in a string of words, then $p_{ij} = P(W_n = W_j | W_{n-1} = W_i)$ is the language model according to Eq. (8.6). The advantage of the word bigram (WB) approach is that the perplexity is reduced considerably (to 20) for the Resource Management task, with essentially no increase in complexity of the implementation.

8.8.1 Control of Word Insertion/Word Deletion Rate

Using a structure of the type shown in Figure 8.9, there is no control on the sentence length. That is, it is possible to generate sentences that are arbitrarily long by inserting a large number of short-function words. To prevent this from occurring, it is a simple matter to incorporate a word insertion penalty into the Viterbi decoding, such that a fixed negative quantity is added to the likelihood score at the end of *each* word arc (i.e., at nodes 5–8 in Figure 8.9). By adjusting the word penalty, we can control the rate of word insertion and word deletion; a very large word penalty will reduce the word insertion rate and increase the word deletion rate, and a very small penalty will have the opposite effect. A value for word penalty is usually experimentally determined to balance these adverse effects.

8.8.2 Task Semantics

We have discussed how task syntax can be incorporated into the overall recognition structure. At the end of this chapter we will briefly describe a general procedure for integrating a semantic component into the recognizer.

8.8.3 System Performance on the Resource Management Task

Using the segmental k -means training algorithm, the set of 47 PLUs of Table 8.1 were trained using a set of 4360 sentences from 109 talkers. The likelihood scores were essentially unchanged after two iterations of the k -means loop. The number of mixtures per state was varied from 1 to 256 in multiples of 2 to investigate the effects of higher acoustic resolution on performance.

To evaluate the recognizer performance, five different sets of test data were used, including:

- | | |
|-----------|---|
| train 109 | A randomly selected set of 2 sentences from each of the 109 training talkers; this set was used to evaluate the ability of the algorithm to recognize the training material |
| feb 89 | A set of 30 sentences from each of 10 talkers, none of whom was in the training set; this set was distributed by DARPA in February of 1989 to evaluate performance |
| oct 89 | A second set of 30 sentences from each of 10 additional talkers, none of whom was in the training set; this set was distributed by DARPA in October of 1989 |
| jun 90 | A set of 120 sentences from each of 4 new talkers, none of whom was in the training set (distributed by DARPA in June of 1990) |
| feb 91 | A set of 30 sentences from each of 10 new talkers, none of whom was in the training set (distributed by DARPA in February of 1991). |

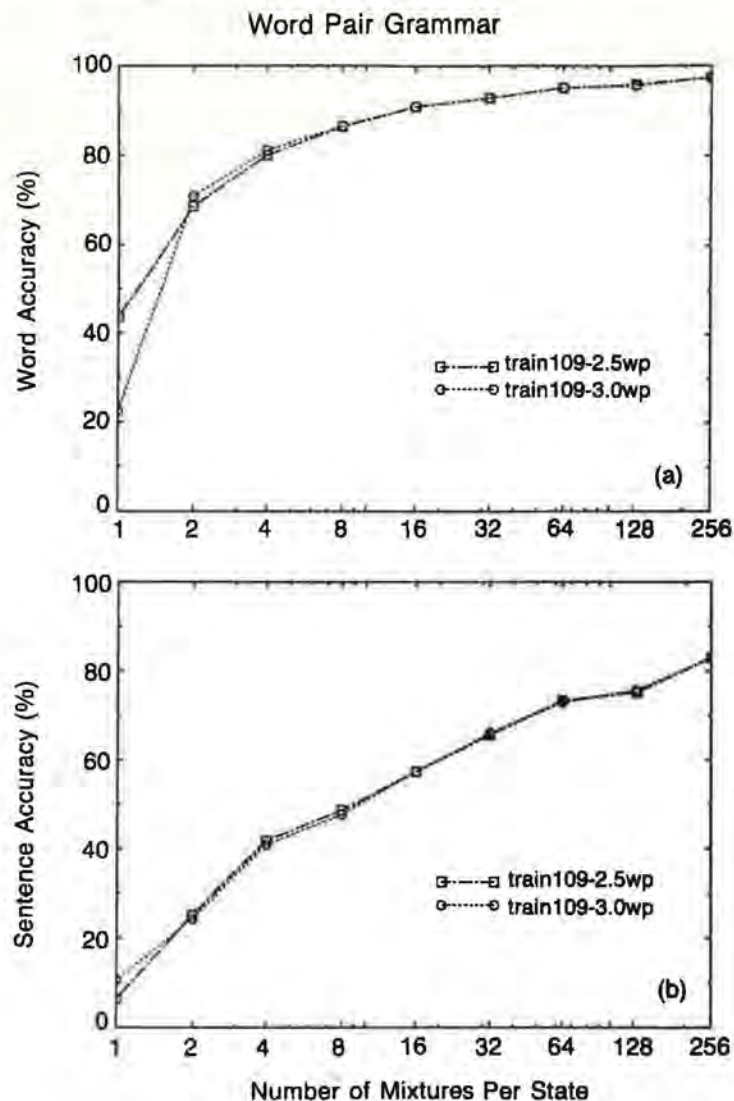


Figure 8.10 Word and sentence accuracies versus number of mixtures per state for the training subset using the WP syntax.

The recognizer performance was evaluated for each of the test sets, using both WP and NG syntax, and with different word penalties. For all cases, evaluations were made using models with from 1 to 256 mixtures per state for each PLU.

The recognition results are presented in terms of word accuracy (percentage words correct minus percentage word insertions) and sentence accuracy as a function of the number of mixtures per state for each PLU model. The alignment of the text of the recognized string with the text of the spoken string was performed using a dynamic programming alignment method as specified by DARPA.

The recognition results on the training subset (train 109) are given in Figures 8.10 (for the WP syntax) and 8.11 (for the NG syntax). The upper curves show word accuracy (in percentage) versus number of mixtures per state (on a logarithmic scale) for two different values of the word penalty, and the lower curves show sentence accuracy for the same

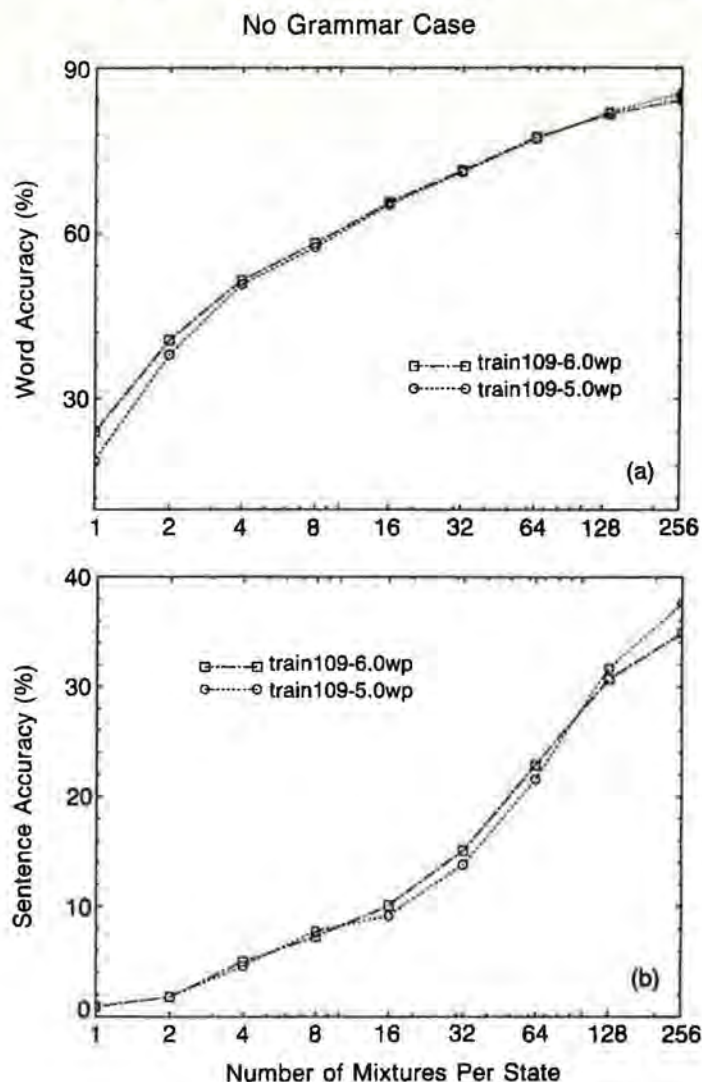


Figure 8.11 Word and sentence accuracies versus number of mixtures per state for the training subset using the NG syntax.

parameters. A sharp and steady increase in accuracy is obtained as the number of mixtures per state increases, going from about 43.6% word accuracy (10.6% sentence accuracy) for 1 mixture per state to 97.3% word accuracy (83% sentence accuracy) for 256 mixtures per state for the WP syntax using a word penalty of 2.5. For the NG syntax (using a word penalty of 6.0), the comparable results were 24% word accuracy (0.9% sentence accuracy) for 1 mixture per state and 84.2% word accuracy (34.9% sentence accuracy) for 256 mixtures per state.

The recognition results on the independent test sets are given in Figures 8.12 (for the WP syntax) and 8.13 (for the NG syntax). Although there are detailed differences in performance among the different test sets (especially for small numbers of mixtures per state), the performance trends are essentially the same for all the test sets. In particular we see that for the WP syntax, the range of word accuracies for 1 mixture per state is 42.9%

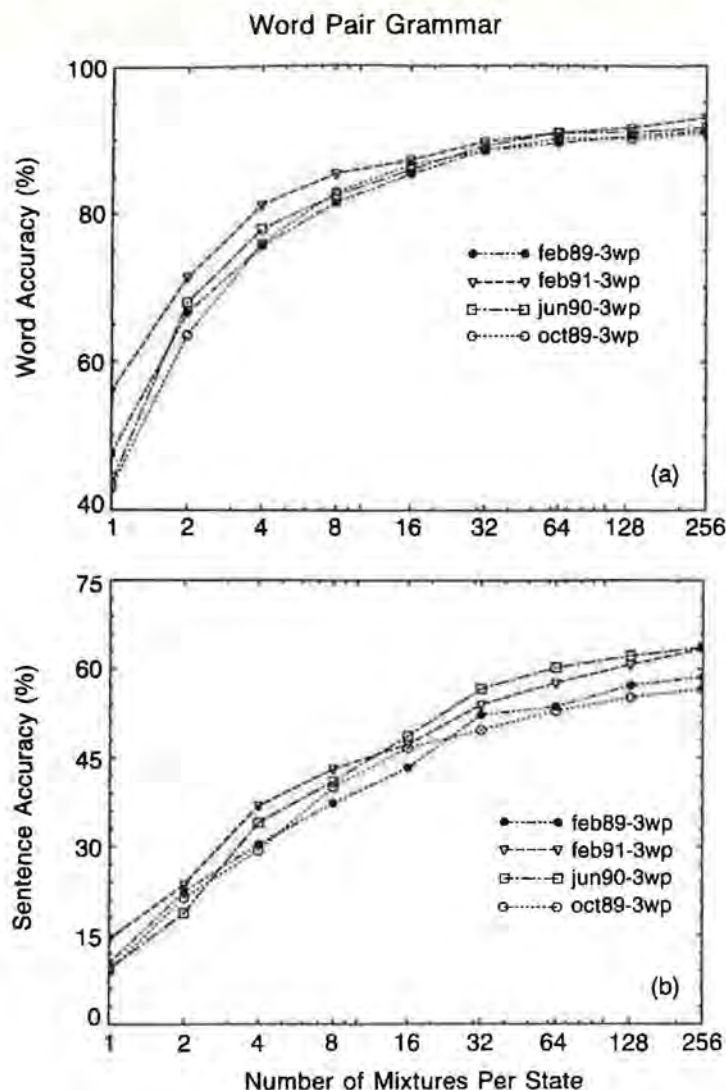


Figure 8.12 Word and sentence accuracies versus number of mixtures per state for the four test sets using the WP syntax.

(for feb 89) to 56.0% (for jun 90), whereas for 256 mixtures per state the range is 90.9% (for feb 89) to 93.0% (for jun 90). For the NG syntax, the range of word accuracies for 1 mixture per state is 20.1% (for feb 91) to 28.5% (for jun 90) and for 256 mixtures per state it is 68.5% (for oct 89) to 70.0% (for feb 91).

Perhaps the most significant aspect of the performance is the difference in accuracies between the test sets and the training subset. Thus there is a gap of 4–7% in word accuracy for the WP syntax at 256 mixtures per state, and a gap of 14.2–15.7% for the NG syntax at 256 mixtures per state. Such gaps are indicative of the ability of the training procedure to overtrain (learn details) on the training set, thereby achieving significantly higher recognition accuracy on this set than on any other representative test set.

The results presented in this section show that a simple set of context-independent PLUs can be trained for a continuous speech large vocabulary recognition task, using

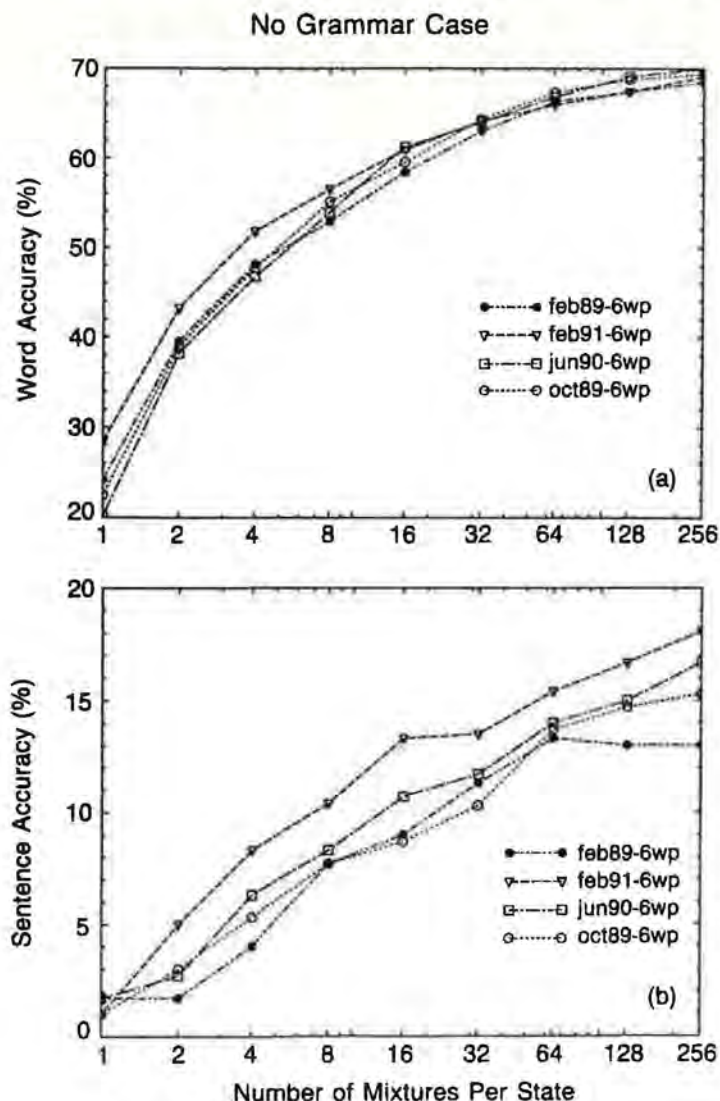


Figure 8.13 Word and sentence accuracies versus number of mixtures per state for the four test sets using the NG syntax.

standard Viterbi training procedures, and be used to provide reasonably good recognition accuracy for a moderately complex task. The key issue now is what can be done, in meaningful ways to improve recognizer performance. To answer this question, we will examine several possible extensions of the basic recognition system in the next few sections.

8.9 CONTEXT-DEPENDENT SUBWORD UNITS

There are several advantages to using a small basic set of context-independent subword units for large vocabulary speech recognition. First of all we have shown that the models of these subword units are easily trained from fluent speech, with essentially no human decisions as to segmentation and labeling of individual sections of speech. Second, the

resulting units are generalizable to new contexts (word vocabularies, tasks with different syntax and semantics) with no extra effort. Finally, the resulting models are relatively insensitive to the details of the context from which the training tokens are extracted. By this we mean that, in theory, we can derive the subword unit model parameters from two arbitrary but sufficiently large training sets of fluent speech (hopefully of the same size and general linguistic content but not necessarily the same vocabulary words and sentences) to obtain essentially the same parameter estimates for each model. In practice, this is almost the case.

However, there are situations in which the subword unit model parameters are extracted from a training set whose linguistic content matches the test set precisely—that is, when the training set is a set of sentences drawn from the recognition task (with the same vocabulary, syntax, and semantics). In such a case, the resulting subword units are somewhat “word sensitive” (showing higher likelihood scores than in the general case) and typically provide higher recognition performance than equivalent model sets derived from arbitrary input speech. In particular, for the Resource Management task discussed in the previous section, “word-sensitive” subword unit models, trained on task-specific training sentences, give about 10% higher word recognition accuracy than the same set of subword unit models trained on arbitrary sentences of comparable size. If the training set is increased in size by a factor of about 3, the word accuracy of the text-independent models approaches that of the word-sensitive models.

Obviously, this performance difference can be attributed to the fact that context-independent subword unit models are not adequate in representing the spectral and temporal properties of the speech unit in all contexts. (By context we mean the effects of the preceding and following sounds as well as the sound stress and information, and even the word in which the sound occurs.) The ultimate effect is a decrease in performance in word and sentence accuracy on speech-recognition tasks.

The solution to this problem is basically a simple, straightforward one—namely, to extend the set of subword units to include context-dependent units (either in addition to or as a replacement for context-independent units) in the recognition system. In theory, the only change necessary in either training or recognition is to modify the word lexicon to be consistent with the final set of subword units. Consider the word “above.” Based on using (1) context-independent units, (2) triphone (left and right context) units, (3) multiple-phone models, and (4) word-dependent units, we could have the following lexical representations:

- | | | | | | |
|------------|---|---------|--------|---------|-------------------------------|
| (1) above: | ax | b | ah | v | Context-Independent Units |
| (2) above: | \$-ax-b | ax-b-ah | b-ah-v | ah-v-\$ | Triphones (Context Dependent) |
| (3) above: | ax2 | b2 | ah1 | v1 | Multiple Phone Units |
| (4) above: | ax (above) b (above) ah (above) v (above) Word-Dependent Units. | | | | |

In representation (2), using triphone units, the number of units needed for all sounds in all words is very large (on the order of 10–20,000). In practice, only a small percentage of such triphone units are used, since most units are seen rarely, if at all, in a finite training set. (We discuss this issue below in more detail.) In representation (3), using multiple models of each subword unit, the idea is to cluster common contexts together so as to reduce the

number of context-dependent models. This leads to problems in defining lexical entries for words. (We discuss this issue further in a later section of this chapter.) Finally, the use of word-dependent units is most effective for modeling short-function words (like a, the, in, of, an, and, or) whose spectral variability is significantly greater than that of long-content words like aboard and battleship. (We discuss the modeling of function words in a later section of this chapter.) Finally, it is both reasonable and meaningful to combine all four types of units in a common structure. In theory, as well as in practice, the training and recognition architectures can handle subword unit sets of arbitrary size and complexity. We now discuss each of these issues in more detail.

8.9.1 Creation of Context-Dependent Diphones and Triphones

Consider the basic set of context-independent PLUs in which we use the symbol p to denote an arbitrary PLU. We can define a set of context-dependent (CD) diphones as

$$\begin{aligned} p_L - p - \$ & \text{ left context (LC) diphone} \\ \$ - p - p_R & \text{ right context (RC) diphone,} \end{aligned}$$

in which p_L is the PLU immediately preceding p (the left context sound), p_R is the PLU immediately following p (the right context sound), and $\$$ denotes a don't care (or don't know) condition.

Similarly we can define a set of context-dependent triphones as

$$p_L - p - p_R \quad \text{left-right context (LRC) triphone.}$$

In theory, the potential number of left (or right) context diphones is 46×45 (for a basic set of 47 PLUs and excluding silence) or about 2070 left context diphone units. The potential number of left-right context triphone units is $45 \times 46 \times 45$ or 93,150 units. In practice, the actual number of context-dependent PLUs actually seen in a finite training set of sentences is significantly smaller than these upper bounds.

To better understand these concepts, consider the RM task (991 word vocabulary) with a training set of 3990 sentences. To use diphone and triphone context-dependent units, we first convert the lexicon to such units using the rule that the initial sound becomes a right context diphone, the middle sounds become left-right context diphones, and the final sound becomes a left context diphone. Hence the word "above" is converted to the set of units $\$-ax-b$, $ax-b-ah$, $b-ah-v$, $ah-v-\$$. (We must use diphone units at the beginnings and ends of words because we do not know the preceding or following words.) The above rule is modified to eliminate triphone middles for words with only two PLUs (e.g., in, or) and to revert to the context-independent PLU for words with only one PLU (e.g., a). Using the above method of creating the lexicon, one can count the number of left-right context-dependent (LRC) units (1778), the number of left-context (LC) units (279), the number of right-context (RC) units (280), and the number of context-independent (CI) units (3) for a total of 2340 PLUs in the training set. This number of units, although significantly smaller than the maximum possible number of context-dependent units, is deceiving because many of the units occur only a small number of times in the training set, and therefore it would be difficult to reliably estimate model parameters for such models.

TABLE 8.4. Number of intra-word CD units as a function of count threshold, T .

Count Threshold (T)	Number of LRC PLUs	Number of LC PLUs	Number of RC PLUs	Number of CI PLUs	Total Number of PLUs
50	378	158	171	47	754
40	461	172	188	47	868
30	639	199	205	47	1090
20	952	212	234	46	1444
10	1302	243	258	44	1847
5	1608	265	270	32	2175
1	1778	279	280	3	2340

To combat the difficulties due to the small number of occurrences of some context-dependent units, one can use one of three strategies. Perhaps the simplest approach is to eliminate all models that don't occur sufficiently often in the training set. More formally we define $c(\cdot)$ as the occurrence count for a given unit. Then, given a threshold T on the required number of occurrences of a unit (for reliable model estimation), a reasonable Unit Reduction Rule is

If $c(p_L - p - p_R) < T$, then

1. $p_L - p - p_R \rightarrow \$ - p - p_R$ if $c(\$ - p - p_R) > T$
2. $p_L - p - p_R \rightarrow p_L - p - \$$ if $c(p_L - p - \$) > T$
3. $p_L - p - p_R \rightarrow \$ - p - \$$ otherwise.

The tests above are made sequentially until one passes and the procedure terminates. To illustrate the sensitivity of the CD PLU set to the threshold T , Table 8.4 shows the counts of LRC PLUs, LC PLUs, RC PLUs, CI PLUs, and the total PLU count for the 3990 sentence training set. For a threshold of 50, which is generally adequate for estimating model parameters, there are only 378 LRC PLUs (almost a 5-to-1 reduction over the number with a count threshold of 1) and a total of 754 PLUs. We will see later that although such CD PLU sets do provide improvements in recognition performance over CI PLU sets, the amount of context dependency achieved is small and alternative techniques are required to create CD PLU sets.

8.9.2 Using Interword Training to Create CD Units

Although the lexical entry for each word uses right or left context diphone units for the first and last sound of each word, both in training and in scoring, one can utilize the known (or postulated) sequence of words to replace these diphone units with the triphone unit appropriate to the words actually (or assumed) spoken. Hence the sentence "Show all ships" would be represented as

\$-sh-ow sh-ow-\$ \$-aw-ℓ aw-ℓ-\$ \$-sh-i sh-i-p i-p-s p-s-\$

using only intraword units, whereas the sentence would be represented as

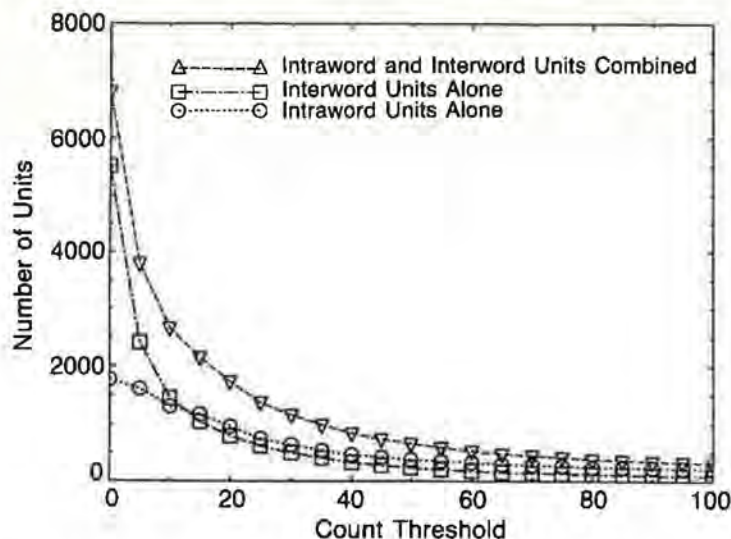


Figure 8.14 Plots of the number of intraword units, interword units, and combined units as a function of the count threshold.

\$-sh-ow sh-ow-aw ow-aw-l aw-l-sh l-sh-i sh-i-p i-p-s p-s-\$

using both intraword and interword units. From this simple example we see that, whereas there were only two triphones based on intraword units, there are six triphones based on intraword and interword units—that is, a threefold increase in context-dependent triphone units. (We are assuming no silence between words; it is straightforward to handle the cases when silence actually occurs between words.) To illustrate this effect, Figure 8.14 shows a plot of the number of intraword units, the number of interword units, and the combined count, as a function of the count threshold, for the 1990 sentence DARPA training set. More than 5000 interword triphone units occur one or more times versus less than 2000 intraword units for the same count threshold.

Even when using interword units, the problems associated with estimating model parameters from a small number of occurrences of the units is the major issue. In the next sections we discuss various ways of smoothing and interpolating context dependent models, created from small numbers of occurrences in the training set, with context-independent models, created from large numbers of occurrences in the training set.

8.9.3 Smoothing and Interpolation of CD PLU Models

As shown above, we are faced with the following problem. For a training set of reasonable size, there is sufficient data to reliably train context-independent unit models. However, as the number of units becomes larger (by including more context dependencies) the amount of data available for each unit decreases and the model estimates become less reliable. Although there is no ideal solution to this problem (short of increasing the amount of training data ad infinitum), a reasonable compromise is to exploit the reliability of the estimates of the higher level (e.g., CI) unit models to smooth or interpolate the estimates of the lower level (CD) unit models. There are many ways in which such smoothing or

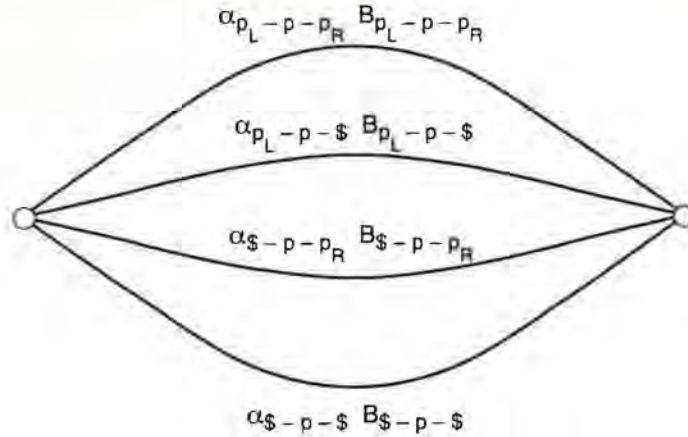


Figure 8.15 Deleted interpolation model for smoothing discrete density models.

interpolation can be achieved.

The simplest way to smooth the parameter estimates for the CD models is to interpolate the spectral parameters with all higher (less context dependency) models that are consistent with the model [12]. By this we mean that the model for the CD unit $p_L - p - p_R$ (call this $\lambda_{p_L-p-p_R}$) should be interpolated with the models for the units $\$ - p - p_R$ ($\lambda_{\$-p-p_R}$), $p_L - p - \$$ (λ_{p_L-p-}) and $\$ - p - \$$ ($\lambda_{\$-p-}$). Such an interpolation of model parameters is meaningful only for discrete densities, within states of the HMM, based on a common codebook. Thus if each model λ is of the form (A, B, π) where B is a discrete density over a common codebook, then we can formulate the interpolation as:

$$\begin{aligned} \hat{B}_{p_L-p-p_R} = & \alpha_{p_L-p-p_R} B_{p_L-p-p_R} + \alpha_{p_L-p-} B_{p_L-p-} \\ & + \alpha_{\$-p-p_R} B_{\$-p-p_R} + \alpha_{\$-p-} B_{\$-p-}, \end{aligned} \quad (8.19)$$

where $\hat{B}_{p_L-p-p_R}$ is the interpolated density. We constrain the α s to add up to 1; hence

$$\alpha_{p_L-p-p_R} + \alpha_{p_L-p-} + \alpha_{\$-p-p_R} + \alpha_{\$-p-} = 1. \quad (8.20)$$

The way in which the α s are determined is according to the deleted interpolation algorithm discussed in Section 6.13. We review the ideas, as they apply to these speech unit models, here. Each of the discrete densities, $B_{p_L-p-p_R}$, B_{p_L-p-} , $B_{\$-p-p_R}$, and $B_{\$-p-}$, is estimated from the training data where a small percentage (e.g., 20%) is withheld (deleted). Using the withheld data, the α s are estimated using a standard forward-backward approach based on the HMM shown in Figure 8.15. The interpretation of the α s is essentially the probability weighted percentage of new data (unseen in training) that favors each of the distributions over the others. Hence, for well-trained detailed models we get $\alpha_{p_L-p-p_R} \rightarrow 1$, whereas for poorly trained models we get $\alpha_{p_L-p-p_R} \rightarrow 0$ (i.e., the LRC model is essentially obtained from interpolating higher-level, lower context dependency models that are better trained than the detailed CD model).

Other smoothing methods include empirical estimates of the α s based on occurrence counts, co-occurrence smoothing based on joint probabilities of pairs of codebook symbols [14], and use of fuzzy VQs in which an input spectral vector is coded into two or more

codebook symbols.

8.9.4 Smoothing and Interpolation of Continuous Densities

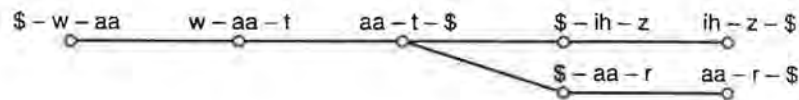
When one uses continuous density modeling of PLUs it is very difficult to devise a good smoothing or interpolation algorithm because the acoustic space of different units is inherently different. There are two reasonable ways to handle this problem. One is to exploit the so called semicontinuous or tied mixture modeling approach discussed earlier in which each PLU uses a fixed set (a codebook) of mixture means and variances, and the only variables are the mixture gains for each model. In this case it is trivial to exploit the method of deleted interpolation on the mixture gains in a manner virtually identical to the one discussed in the previous section.

An alternative modeling approach, and one more in line with independent continuous density modeling of different sounds, is to use a tied-mixture approach on the CI unit level; that is, we design a separate (large) codebook of densities for each CI PLU and then constrain each derived CD unit to use the same mixture means and variances but with independent mixture gains. Again we can use the method of deleted interpolation to smooth mixture gains in an optimal manner.

8.9.5 Implementation Issues Using CD Units

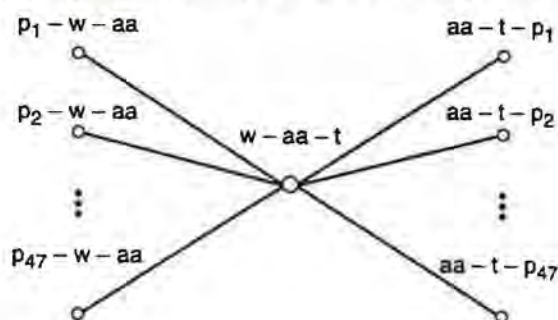
The FSN structure of Figure 8.9 is used to implement the continuous speech-recognition algorithm based on a given vocabulary and task syntax ([15–19]). The structure is straightforward to implement when using strictly intraword units because there is no effect of context at word boundaries. Hence the models (HMMs) for each word can be constructed independently and concatenated at the appropriate point of the processing. This is illustrated below for the recognition of the string “what {is, are}” based on intraword units, where the individual words are represented in the lexicon as

what — { $\$-w-aa$, $w-aa-t$, $aa-t-\$$ }
 is — { $\$-ih-z$, $ih-z-\$$ }
 are — { $\$-aa-r$, $aa-r-\$$ }

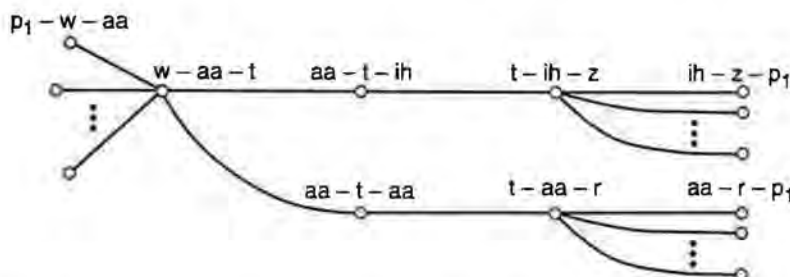


(If we allow silence between words there is a trivial modification to include a silence node after the word /what/.) When we include interword units in the recognition stage the FSN becomes considerably more complicated because the first unit of each word (which we call the head unit) is variable depending on the last unit of each possible preceding word; similarly, the last unit of each word (which we call the tail unit) is variable depending on the first unit of each possible following word. (We call the set of units between the head unit and the tail unit, the body units.) Thus, in theory, a word like “what” consists of (up

to 47 head units and (up to) 47 tail units, and would be represented as



In practice many, if not most, of the head as well as tail units don't exist; hence, the structure is generally considerably less complex. Thus the FSN network of the strings "what {is, are}" becomes



that is, a considerably more complex network results. (Interestingly, the inclusion of silence adds only a single extra path to each branch of the network.) The bookkeeping associated with such networks can easily get out of hand and dominate the overall computation. Fortunately, several network architectures have been devised for efficiently handling the bookkeeping associated with such networks [15]. Interesting special cases occur when the number of units within a word falls below three. When there are exactly two units in a word, there are no body units so the variable head units merge with the variable tail units. When there is only a single unit within a word, there are no body units nor is there a tail unit. Effectively, the bookkeeping must look at both the preceding word set of tail units and the following word set of head units to handle this case. The three cases described above—namely, implementations of words with three or more units, words with two units, and words with one unit—are illustrated in Figure 8.16.

8.9.5.1 Word Junction Effects

The assumption that is made when training interword units is that in continuous speech, words are pronounced similar to the way they are pronounced in isolation. In most cases this assumption is reasonable in that the coarticulation phenomena at word boundaries only lead to small (soft) changes in the word pronunciation and therefore can readily be modeled by interword units based on the concatenation of the tail unit from one word with the appropriate head unit from the following word. However, in some cases, the pronunciation changes are radical (hard) changes in which a boundary sound (tail or head)

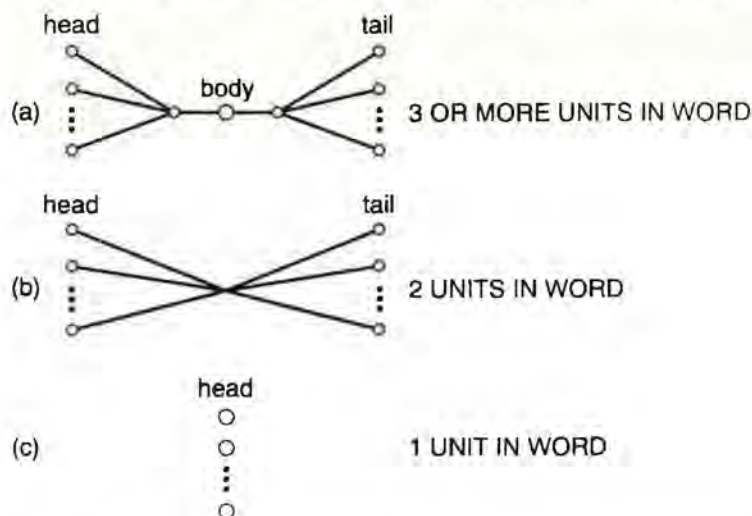


Figure 8.16 FSN representation of words with three or more units (a), two units (b), and one unit (c).

is completely deleted or replaced with a totally different sound. Examples of such hard changes include the strings "what time" and "did you," among others. In the string "what time," the double stop consonant (t followed by t) between words is replaced with a single occurrence of t; hence one of the *ts* is deleted. In the string "did you," the standard phonetic transcription would be /d ih d y uw/; however, in continuous speech the actual transcription would be /d ih j h uw/ (or even /d ih jh ax/) where the /d y/ boundary phones are changed to the single sound /jh/.

Phonologically predictable changes of the type shown above (the so-called hard sound changes) cannot easily be learned from the training procedure because they occur infrequently and they lead to radically different sounds than would be predicted from the concatenation of boundary sounds of the relevant words as spoken in isolation. To handle these hard changes correctly, a set of phonological rules has to be superimposed on both the training and recognition networks (in a relatively straightforward (brute force) manner). There are about 11 such rules that handle most of the known phonological changes in English [16].

Some typical phonological rules include the following:

- Rule 1** Geminate deletion. If a word ends in a consonant and the following word begins with the same consonant, the ending consonant is deleted; e.g., the final /t/ is deleted in the pair "what time."
- Rule 2** Palatization. If a word ends in a /d/, and the following word begins with a /y/, then (optionally), the final sound can be converted to a /jh/, and the initial sound of the following word is deleted. Thus the words "did you" can be spoken as either /d ih d y uw/ or /d ih jh uw/.
- Rule 3** Plosive deletion. If a word ends in the nasal /n/ followed by a plosive sound, and the following word begins in a plosive sound, then the final plosive in the initial word is deleted. Thus the words "went down" can be spoken as /w eh n d aw n/.

The complete set of rules is available in Reference [15].

8.9.5.2 Variance Estimation Problems

Perhaps the most difficult problem, in training, when using continuous density HMMs, is the estimation of mixture variances when the amount of training data is small (as is almost always the case when using context-dependent units). The problem is that to maximize the likelihood on the training data, the estimation procedure often tries to make the variance very small (i.e., choosing data samples that are very close to each other in value). Although this leads to good training likelihood scores, it often provides poor matches to independent test data. Hence, some protection against variances getting too small in training is required.

Several proposals have been made as to how to realistically control the variance of the estimates to prevent such effects from occurring. One simple one is to tie variances across units, states, and even words, ultimately leading to a grand variance for each spectral component that is independent of the unit, state, and word. This idea is reasonable and has been shown to work well in practice [20]. An alternative is to set a floor on the variance of each spectral component that is based on a statistical analysis of the range of values of the variance component for different sounds, states, etc. Thus rather than using a grand variance, the concept of setting a variance clipping threshold at an appropriate point of the distribution (e.g., 2 sigma below the mean) preserves the (reasonable) range of variance estimates while at the same time preventing the variance from getting unreasonably small. (One could also argue that a high clipping threshold would prevent the variance from getting unreasonably large; in practice, the process of maximizing likelihood prevents this from happening. See Section 6.5 for a more complete discussion of these concepts.)

8.9.6 Recognition Results Using CD Units

A key issue in continuous speech recognition is the total number of subword units used in the system. We have already discussed several different types of subword units, including context-independent units, intraword context-dependent units, interword context-dependent units, and various combinations of these. In later sections of this chapter we will extend the unit classes to include position-dependent units, function word-dependent units, and even function phase-dependent units.

On the one hand, it seems clear that as we add more units with greater context dependency, the performance of the recognition should continue to improve. On the other hand, for a fixed training set, the amount of training data available for estimating model parameters of context-dependent units becomes smaller as the number of units increases. Hence the reliability of the estimates of model parameters decreases and therefore recognition performance falls. (To combat this second effect, various smoothing and interpolation procedures have been devised.) The overall result is that recognition performance is maximized for a finite size subword unit set whose size depends on the training data, the recognition vocabulary, the task syntax, and the method for creating the context-dependent units. In this section we present several results illustrating this trade-off between number of subword units and overall word accuracy of the recognizer.

TABLE 8.5. Word error rates as a function of occurrence threshold for the feb 89 test set using intraword units with a 38 component/vector analysis.

Threshold	∞	30 ^a	25 ^a	20 ^a	15 ^a	10 ^a
Number of Units	47	1090	1215	1444	1694	1874
WP Word Error (%)	14.0	6.7	7.0	7.1	7.4	7.6
NG Word Error (%)	40.0	25.0	24.8	25.0	25.2	25.6

^aSixteen mixtures per state were used for these model sets.

TABLE 8.6. Word error rates as a function of occurrence threshold for the feb 89 test set using both intraword and interword units (independently) with a 38 component/vector analysis.

Threshold	∞	30 ^a	25 ^a	20 ^a	15 ^a	10 ^a
Number of Units	47	1769	2125	2534	2985	3863
WP Word Error (%)	9.1	4.6	4.7	4.6	4.7	5.3
NG Word Error (%)	32.7	20.8	19.8	19.4	20.6	20.9

^aSixteen mixtures per state were used for these model sets.

For evaluating speech-recognition performance, we use the feb 89 test set of 300 sentences spoken by 10 adult male and female talkers (30 sentences per talker). Using, as a baseline system, the recognizer based on the 47 context-independent units with 256 mixtures per state, the unit reduction rule was used at several thresholds to generate unit sets with up to 1874 intraword units (no interword units were used here), and the tests were done using cepstral plus differential cepstral (delta and delta-delta) with differential energy (first and second order) parameters (38/vector) with both the word pair (WP) and no grammar (NG) syntaxes. The word recognition accuracies for these systems are given in Table 8.5. A significant improvement in performance is achieved when adding context-dependent intraword units (e.g., from 14% to 9.2% error rate for the WP case); however, increasing the total number of units from 638 to 915 or 1759 or 2340 does not reduce word error rate but instead increases it slightly. This is the tradeoff referred to above. For the NG syntax a similar trend is observed, although the recognizer performance is relatively flat for a large range of units.

The results when using both intraword and interword units (independently, see next section) are shown in Table 8.6. For this test, we again used the feb 89 test set; however, we used the full 38 component/vector analysis frame (including delta-delta cepstral values, delta energy, and delta-delta energy). The effects of the enhanced analysis frame are seen in the improved performance of the 47 PLU set where the error rate falls from 14% to 9.1% for the WP syntax, and from 40% to 32.7% for the NG syntax. Similarly the use of interword units (along with the enhanced analysis) reduced the error rate to 4.6% for the WP case (using 1769 units) and to 19.4% for the NG case (using 2534 units). Again

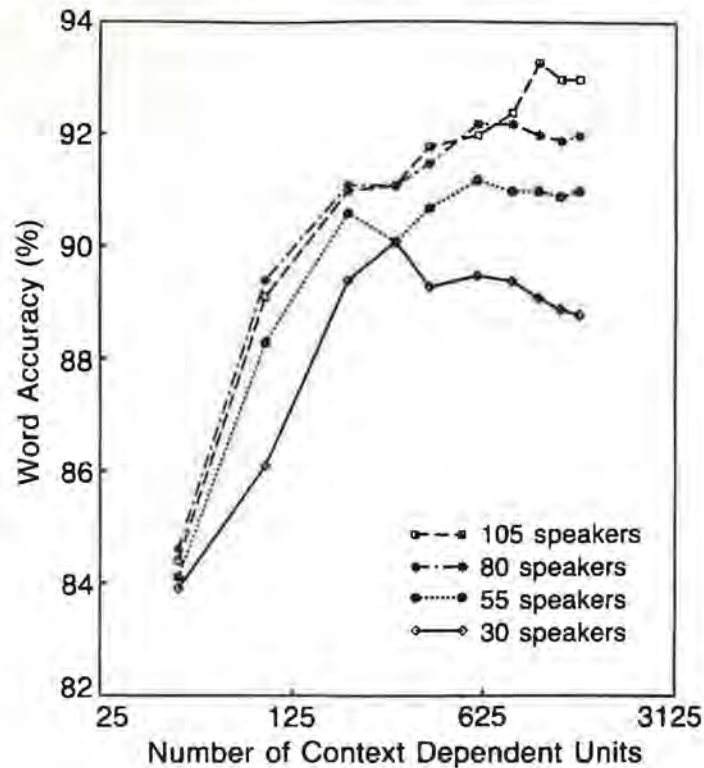


Figure 8.17 Word accuracy (%) as a function of the number of generalized triphone models for several training set sizes (after Lee [2]).

we clearly see the saturation in performance as the number of units increases due to the problems in reliably estimating parameters of the context-dependent models from a finite training set.

To illustrate the effects of training set size on recognition performance even more dramatically, Figure 8.17 (due to K. F. Lee) shows a plot of word accuracy versus number of units (called generalized triphone models in Lee's notation) for different size training sets (measured in terms of the number of speakers in the set). For the smallest training set (30 speakers), the word accuracy peaks at around 300 units and then falls dramatically beyond this point. For the 55-speaker training set the word accuracy peaks at around 625 units and then falls slightly. For the 80 and 105 speaker sets the performance peaks at about 1000 and 2000 units. These results dramatically illustrate the difficulties in creating subword unit sets with a large number of context-dependent units.

8.9.7 Position Dependent Units

When using both intraword and interword units, it is natural and reasonable to combine occurrences of the same unit independent of whether they occurred within the word or across words. It has been observed that phones within words are significantly more stable, acoustically, than phones occurring at word boundaries. Thus it seems plausible that the spectral behavior of the same intraword and interword unit could be considerably different.

To illustrate this point, two sets of context-dependent subword units were created using the same unit reduction rule [21]. In one set common occurrences of intraword and interword units were combined; in the other set they were modeled independently according to their positions within the words or across words (thus the name position dependent). Using a threshold of 30, there were 1282 combined units, including 1101 left-right context units, 99 left-context units, 35 right-context units, and 47 CI units, and 1769 separate position-dependent units including 913 intraword units and 856 interword units.

To show that the spectral properties of these two sets were different, the histograms of unit separation distances for the two sets were computed as follows. For each unit, λ_p , in each set, we computed the minimum distance (likelihood separation) as

$$D(p) = \min_{q \neq p} \{L(Y_p | \lambda_p) - L(Y_p | \lambda_q)\}$$

where Y_p represents the training data segments used to estimate λ_p . $D(p)$ represents the smallest likelihood score difference when using any other model than the one created from Y_p . (In practice, the computation is performed only for models, λ_q , which had the same base unit as λ_p since all other models gave significantly larger difference scores.) The histograms of unit separation distance for the 1282 PLU set of combined intraword and interword units, and for the 1769 PLU set of position dependent units are shown in Figure 8.18. For the 1769 PLU set almost *all* of the unit separation distances are larger than 2.0 (difference in log likelihoods), including the cases where the same unit occurred in both intraword and interword contexts. The average unit separation distance for this set is about 9.0. For the 1282 PLU set the histogram is skewed to the left, showing many small unit separations, with an average distance on the order of 4–5. The results clearly show that the spectral properties of context-dependent units are often significantly different within words than when they occur at word boundaries.

8.9.8 Unit Splitting and Clustering

We have shown in previous sections that it is relatively simple to train models for a small set of context-independent units from a training set of a reasonable size. The problem is that the recognizer performance is not good enough for large vocabulary continuous speech-recognition tasks. We also discussed one simple way to train models for a large set of context-dependent units from the same training set. Here the problem is the inadequacy of training data, which leads to poor estimates of model parameters for all but a small subset of the units observed in a typical training set. The result of the poor model estimates is that recognition performance saturates for about 1000–2000 context-dependent units and either remains constant or decreases as the number of units trained increases.

Thus a key issue in the design and implementation of large vocabulary continuous speech recognizers is how to efficiently determine the number and character of the context-dependent units that give best recognition performance for a given training set. Unfortunately, there is no simple answer to this question. In this section we discuss several proposed methods based on the concepts of either starting from a small set of context-independent units and iteratively splitting the units, or of starting from a large set

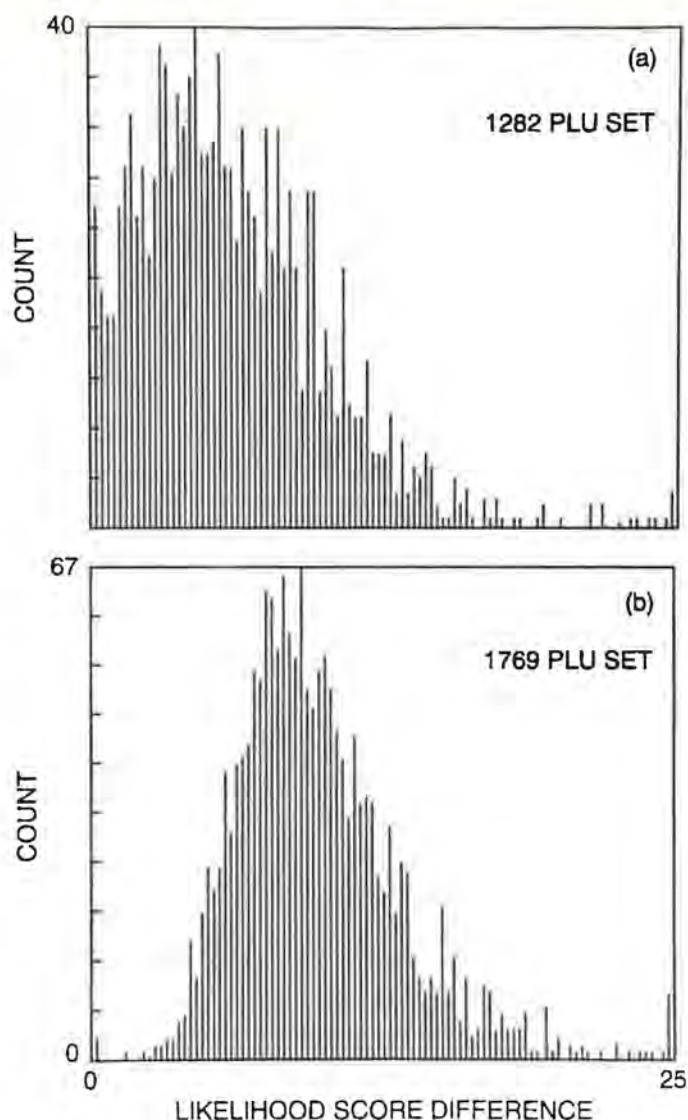


Figure 8.18 Histograms of the unit separation distance for (a) combined intraword and interword units (1282 PLUs) and (b) separate intraword and interword units (1769 PLUs) (after Lee et al. [19]).

of context-dependent units and merging similar units to reduce the number of units based on some type of clustering procedure.

8.9.8.1 Splitting of Subword Units

The basic idea of subword unit splitting is illustrated in Figure 8.19. We assume that for each subword unit p_i (with model λ_i), representing a context-independent unit, there is some inherent internal distribution of training tokens that naturally clusters into two or more clusters. (Within the figure we show three clusters, namely p_i^1 , p_i^2 , and p_i^3 .) The clusters represent classes of sounds that are all labeled as p_i , but which have different spectral

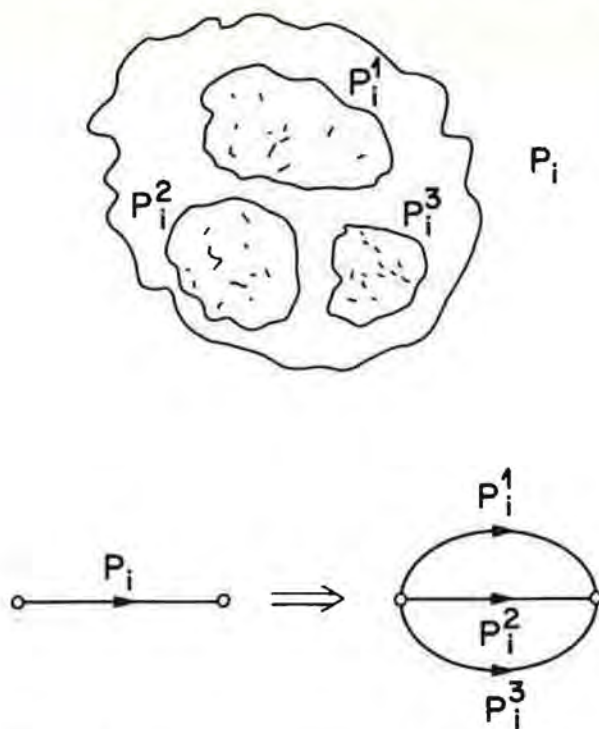


Figure 8.19 Splitting of subword unit p_i into three clusters (after Lee et al. [7]).

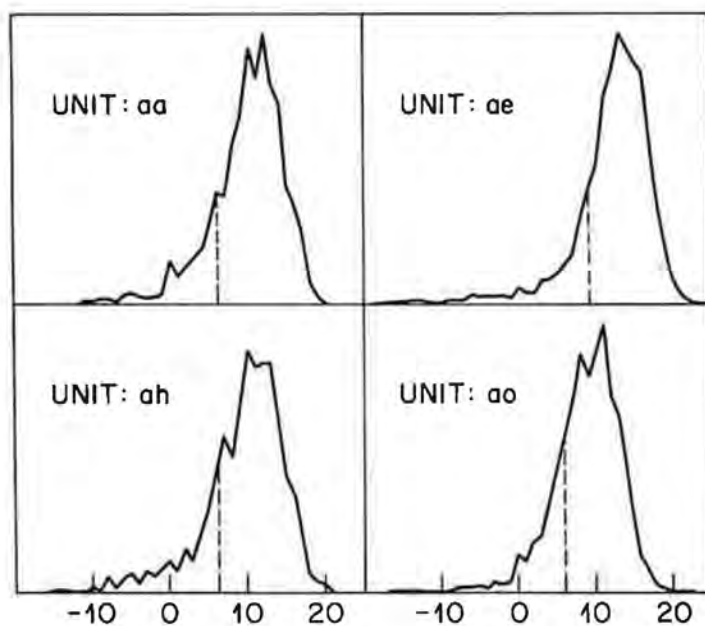


Figure 8.20 Histograms of likelihood score for four context-independent units (vowels) (after Lee et al. [7]).

properties depending on the context in which they occur. Once the separation of p_i into clusters is achieved, we have effectively created multiple models of the context-independent subword unit, as shown at the bottom of Figure 8.19.

There are several ways to create the clusters for each unit, but a particularly simple (and meaningful) one is based on the following argument. If we examine the histogram of (log) likelihood scores for each training token which is labeled p_i we get curves similar to those shown in Figure 8.20. The likelihood score histograms show that for a large percentage of the training tokens, good scores are obtained using the context-independent unit. These training tokens are (relatively) well represented by λ_i and do not need to be split off. Instead the low tail of the histogram (below the dashed lines) represents training tokens whose likelihood scores are relatively low and these tokens need an alternative representation (model) to be well represented.

Based on the above discussion, a simple procedure for splitting off training tokens with low likelihood scores and creating a new model from these tokens is as follows:

1. For each subword unit, p_i , which is to be split (not every unit need be split), all training tokens whose likelihood scores fall below a threshold are split off and used to estimate an additional model for that unit.
2. The segmental k -means training procedure is iterated on the split-off tokens until the new model reaches convergence.
3. The above procedure (steps 1 and 2) is iterated until the desired number of models, for each subword unit, is obtained.

The results (in terms of average likelihood score over the entire set of units) of applying the above splitting procedure to the 47 PLU set of context-independent units is shown in Figure 8.21. The results are shown, as a function of iteration number, for splitting each of the 47 models into 2, 3, and 4 models. The procedure converges rapidly and provides small but consistent increases in average likelihood scores.

The above model splitting procedure leads to one major difficulty, namely, How do we modify the word pronunciation dictionary to account for the presence of multiple versions of each subword unit? The inherent problem is illustrated in Figure 8.22, which shows the networks for a complete set of word models assuming every version of each sound in the word can follow every version of every other sound in the word (part a), or that instead we determine one or two best representations of each word via some type of word learning procedure (part b). The problem with the network of part a is that a word with N sounds (e.g., "often" has three sounds, /ao, f, en/) has 2^N representations when we use the complete network (e.g., eight versions of "often") with two models for each sound. This means not only more computation, but even worse, more chances to cause word insertion or substitution in the recognition phase because there are so many more ways in which the words can occur. The network of part b, in which we explicitly enumerate the version of each unit used for each word, is far more viable; however, the problem is how to estimate the best sequence of units for each word in the lexicon. To do this properly we need occurrences, within the training set, of each word in the vocabulary from which we use the network of part a and backtrack to get the best sequences of the type shown in part b.

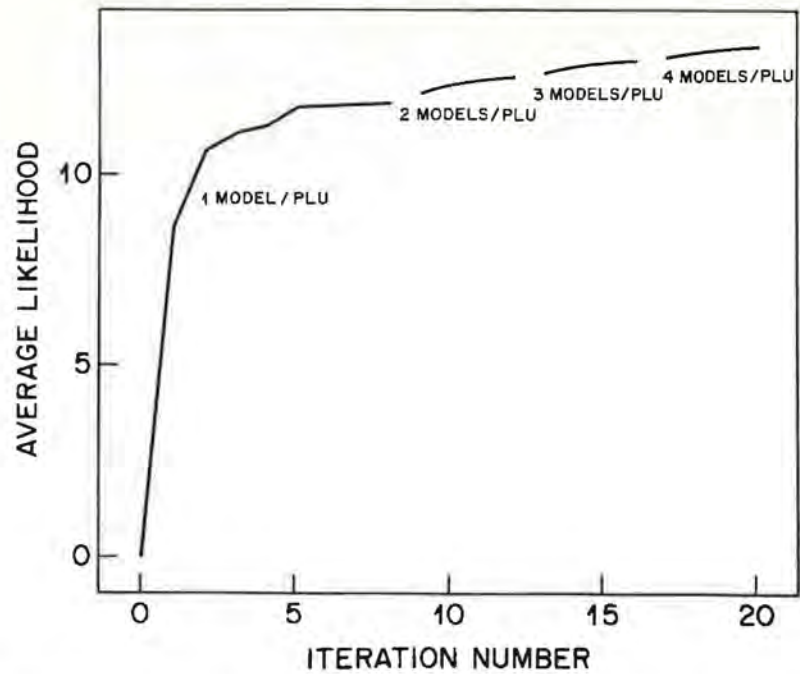


Figure 8.21 Average likelihood scores for sets of PLU models obtained from model splitting (after Lee et al. [7]).

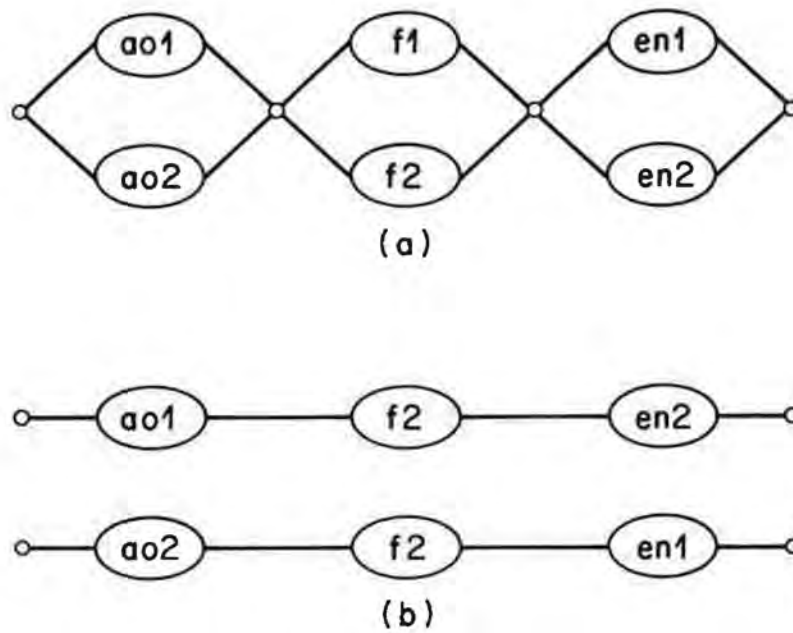


Figure 8.22 Word networks based on all combinations of units (a), or selected combinations of units (b) (after Lee et al. [7]).

For words that don't occur in the training set, some canonic representation must be relied on (e.g., use the primary model for each sound as a default). The necessity of having each word occur in the training makes this type of splitting method less viable than alternative procedures.

8.9.8.2 Clustering of Context Dependent Units

The alternative to model splitting (a top-down approach) is model clustering (a bottom-up approach) in which we initially start with the complete set of context-dependent units (as many as exist in the training set based on a count threshold of 1) and then sequentially merge units (actually the training tokens associated with the units) so that the decrease in likelihood score is minimized at each step. (In practice, we merge only units whose contexts are comparable, e.g., unit $p_i - p_j - p_k$ could be merged with units $p_i - p_j - p_\ell$, $\ell \neq k$, or $p_\ell - p_j - p_k$, or $\$ - p_j - p_k$, or $p_i - p_j - \$$, or $\$ - p_j - \$$.) This procedure is iterated either until a desired number of units is reached or until the resulting decrease in likelihoods gets too large.

A key advantage to model clustering is that it is trivial to modify the word lexicon to account for the decrease in units from merging. The procedure is basically to change each occurrence of both $p_i - p_j - p_k$ and $p_\ell - p_j - p_n$ to the merged unit, call it $\tilde{p}_i - p_j - p_k$, whenever they occur in the lexicon. Thus model clustering is inherently simpler to implement than model splitting and therefore has been used more widely in practical systems.

Many variations on model clustering have been proposed, including knowledge-based allophonic clustering [22], in which specific knowledge of the vowel and consonant contexts is explicitly used to guide the clustering procedure, and CART-based phonetic clusters in which a decision tree is used to choose the most reasonable clustering sequence based on phonetic considerations.

8.9.9 Other Factors for Creating Additional Subword Units

In practice, the training methods for creating robust, complete sets of subword unit models for representing continuous speech are up against hard physical limits, including amount of training data and ability to reliably estimate model parameters from insufficient training. To obtain improvements in recognition performance, subject to the above constraints, several ideas have emerged for creating specialized units and models. For completeness, we briefly outline several interesting proposals that have been advanced along these lines.

A key source of difficulty in continuous speech recognition are the so-called function words, which include words like *a*, *and*, *for*, *in*, and *is*. These function words have the following properties:

1. They are generally unstressed in speech.
2. They are poorly articulated in continuous speech.
3. They are highly variable in pronunciation depending on context.

4. They account for a large percentage of the word recognition errors in continuous speech (upward of 50–70% in some tests).

To combat these problems, one simple idea is to represent function words independently of the rest of the training set, using either whole-word models, multiple pronunciations in the lexicon (e.g., *the*, *thee*), or special subword units, called function word dependent units, trained directly from occurrences of the function words within the training set. Experience shows small but consistent improvements in recognition performance when function word dependent units are added to the standard set of subword units.

The idea of representing function words can be extended to the representation of function phrases such as “in the,” or “what is.” Thus, specialized units can be created for these combinations in much the same way as for individual function words. Again there are small performance gains that are achieved when using function phrase units.

Another interesting idea is to create separate sets of units for both male and female talkers. The idea is that the spectral properties of the units are distinct for males and females. The problem is that by separating male from female talkers, the amount of training data for each separate gender set is reduced. Hence the reliability of the estimates of both sets of models is reduced even further. Experience again shows small, consistent gains in recognition performance using gender-specific models; hence this method is worth considering for practical implementations.

Finally it has been proposed that a combination of word models and subword unit models might give the best performance for specific tasks. The idea is that for words that do occur often in the training set (e.g., function words), creation of whole-word models provides the highest recognition performance. For all other words in the lexicon, some type of subword units is required. Hence a combination of word and subword units would probably lead to the best implementation for many applications. This idea has yet to be evaluated in a practical application.

8.9.10 Acoustic Segment Units

In this chapter we have shown that large vocabulary continuous speech recognition systems use a combination of ideas from phonetics and acoustics to define subword units and to create a “consistent” framework for training the units and implementing the overall recognition structure. The resulting system is neither phonetically nor acoustically consistent, but is instead a hybrid of the two methodologies. This is why the resulting subword units are called phonelike units (PLUs) rather than phones or allophones.

In an attempt to create a consistent acoustic framework (devoid, in theory, of the phonetic basis), it is possible to define a set of acoustic segment units (ASUs) that can be trained from continuous (unlabeled) speech, and which form a basis for representing any spoken input. In concept, all one need do is to have a procedure that automatically segments fluent speech into unlabeled sections [9, 23] (based on a maximum likelihood procedure using some type of spectral similarity measure), and then cluster the resulting segments to create a codebook of ASUs.

The problem now becomes one of creating an acoustic lexicon that represents words in the recognition vocabulary in terms of the appropriate sequence of ASUs. For systems in which every vocabulary word is seen in the training set, techniques for creating the acoustic lexicon exist and appear to work well [9, 24]. However, for large vocabulary systems the problem of automatically creating the acoustic lexicon remains a major obstacle to the practical use of ASUs.

8.10 CREATION OF VOCABULARY-INDEPENDENT UNITS

A major limitation in the training procedures discussed in this chapter is that the resulting subword unit models are not truly vocabulary independent. This is because the unit models are generally trained from tokens that occur in only a small subset of the possible contexts, and this subset is from the same words as used in the recognition tests. As such, the resulting units are word/vocabulary dependent and do not perform well for tasks in which different vocabularies and task syntaxes are used.

To alleviate this problem of vocabulary dependence, the "ideal" training procedure would be to use a training set that is completely independent of the test material, both in vocabulary and in syntax. If a sufficiently large training set is available, the units models will eventually converge so that the resulting recognition performance is virtually independent of the vocabulary and task.

To evaluate this idea, two experiments were run at CMU [25]. Using a vocabulary-independent training set of 15,000 sentences, subword unit models were created from subsets of 5000 (VI-5000), 10,000 (VI-10000), and 15,000 (VI-15000) sentences and tested against two tasks, namely a 122-word office correspondence task, and the 911 RM task. The results of these two experiments are given in Tables 8.7 and 8.8. For comparison, in Table 8.7, results based on training models from 1000 sentences from the office correspondence task (VD-1000) are also given. For 5 times the size training set the error rate from VI-5000 models is more than twice that of the VD-1000 models. Even with 15 times as much training data, the error rate is still somewhat larger for the VI-15000 model than for the VD-1000 model.

An even worse performance is seen for the RM task in which a VI-15000 training set led to almost twice the error rate of the RM-4200 (sentence) training set. (A final test was run at CMU in which new test sentences were recorded at CMU under the same recording conditions as those of the VI-15000 set, and the resulting recognition performance of both the VI-15000 set and the RM-4200 set were comparable. Thus, some of the large differences in performance result from differences in recording conditions.)

The results of these tests show that robust techniques for creating truly vocabulary-independent units are yet to be devised. Until such methods are available, truly continuous speech recognition for unlimited vocabularies and tasks will be out of range. The interim solution is to use VI models and bootstrap them to VD models for specific applications. Experimental evidence exists that such procedures are viable for many applications.

TABLE 8.7. Recognition performance on 122-word, office correspondence task with both VI and VD models (after Hon & Lee [25]).

Training Set	Word Coverage (%)	Triphone Coverage (%)	Word Error Rate (%)
VI-5000	44.3	63.7	23.9
VI-10000	63.9	95.3	15.2
VI-15000	70.5	99.2	13.3
VD-1000	100	100	11.4

TABLE 8.8. Recognition performance on 991-word, RM task, with both VI and VD models (after Hon & Lee [25]).

Training Set	Word Coverage (%)	Triphone Coverage (%)	Word Error Rate (%)
VI-15000	57.0	90	15.4
RM-4200	100	100	8.3

8.11 SEMANTIC POSTPROCESSOR FOR RECOGNITION

The final stage of processing in most speech recognizers is a semantic processor whose job is to eliminate from consideration all semantically meaningless sentences. In a sense, the semantic processor exploits the fact that the syntax used in recognition has a great deal of overcoverage; that is, it allows meaningless sentences to be passed to the semantic analyzer. The semantic processor can use the actual perplexity of the task (generally much lower than the perplexity of the syntax) to convert the recognized output to a semantically valid string.

In theory, the semantic processor should be able to communicate back to the recognizer to request a new string whenever the resulting string is deemed invalid. In practice, one of two simple strategies can be used; either the recognizer can generate a list of the best N sentences ($N = 500 - 1000$) that the semantic processor can search until a valid one is found, or it can assume that the best (recognized) string is semantically "close" to the correct string and therefore process it appropriately to determine a valid approximation.

Rather than discussing the details of how such semantic processing is done in practice, Figure 8.23 shows plots of improvements in word and sentence accuracy for different sets of subword units due to the use of a simple semantic postprocessor for the RM task [26]. Improvements in word accuracy of up 10% and improvements in sentence accuracy of over 20% are achieved, even with simple processing.

8.12 SUMMARY

The framework of large vocabulary, continuous speech recognition is well established.

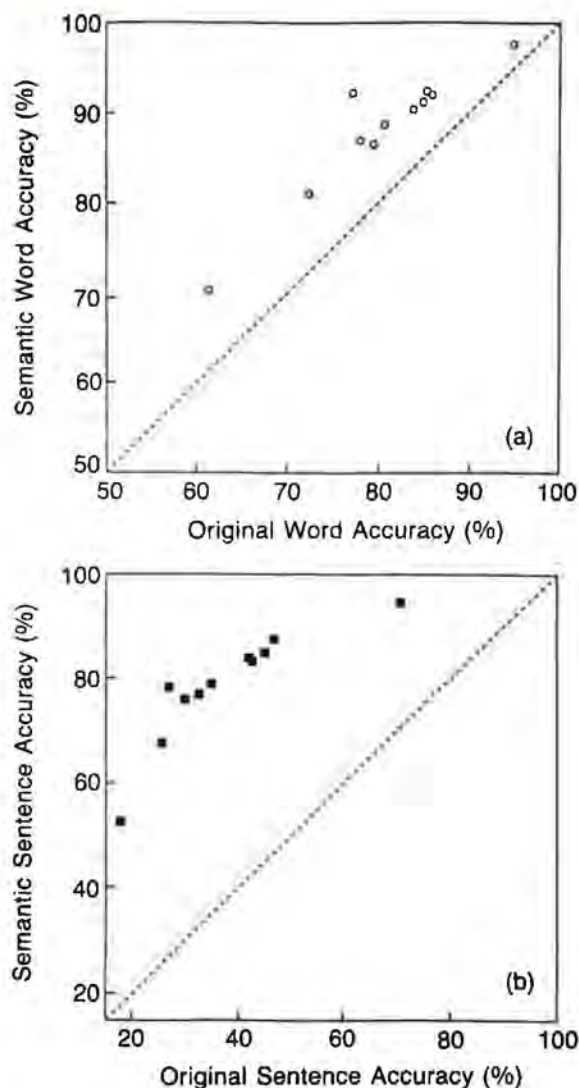


Figure 8.23 Word and sentence accuracy improvements in RM after semantic processing (after Pieraccini and Lee [26]).

Techniques for training subword models have been developed and work well in practice. Recognition systems have been developed and these, too, work well in practice. Recognition systems have been implemented with upward of 1000–20,000 word vocabularies using upward of 1000–2000 subword units. Many unanswered questions remain. A key one is how to efficiently choose and design context-dependent, vocabulary-independent units from training sets of reasonable (but finite) size. Other issues concern effectiveness of different spectral representations, including codebooks and tied-mixture densities, efficiency of implementation of search strategies, and efficient implementations of task syntax. Finally, the issues involved with task semantics are yet to be fully understood or resolved. Large vocabulary recognition has come a long way, but a great deal remains to be done before such systems will be used for practical applications.

REFERENCES

- [1] F. Jelinek, "The Development of an Experimental Discrete Dictation Recognizer," *Proc. IEEE*, 73: 1616–1624, November 1985.
- [2] K.F. Lee, *Automatic Speech Recognition—The Development of the SPHINX System*, Kluwer Academic Publishers, Boston, 1989.
- [3] D.B. Paul, "The Lincoln Robust Continuous Speech Recognizer," *Proc. ICASSP 89*, Glasgow, Scotland, pp. 449–452, May 1989.
- [4] R. Schwartz et al., "The BBN BYBLOS Continuous Speech Recognition System," *Proc. DARPA Speech and Natural Language Workshop*, pp. 94–99, February 1989.
- [5] M. Weintraub et al., "Linguistic Constraints in Hidden Markov Model Based Speech Recognition," *Proc. ICASSP 89*, Glasgow, Scotland, pp. 699–702, May 1989.
- [6] V. Zue, J. Glass, M. Phillips, and S. Seneff, "The MIT Summit Speech Recognition System: A Progress Report," *Proc. DARPA Speech and Natural Language Workshop*, pp. 179–189, February 1989.
- [7] C.H. Lee, L.R. Rabiner, R. Pieraccini, and J.G. Wilpon, "Acoustic Modeling for Large Vocabulary Speech Recognition," *Computer Speech and Language*, 4: 1237–165, January 1990.
- [8] A.E. Rosenberg, L.R. Rabiner, J.G. Wilpon, and D. Kahn, "Demisyllable Based Isolated Word Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, ASSP-31: 713–726, June 1983.
- [9] C.H. Lee, F.K. Soong, and B.H. Juang, "A Segment Model Based Approach to Speech Recognition," *Proc. ICASSP 88*, New York, pp. 501–504, April 1988.
- [10] J.R. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Models for Large Vocabulary Isolated Speech Recognition," *Proc. ICASSP 89*, Glasgow, Scotland, pp. 13–16, May 1989.
- [11] X.D. Huang and M.A. Jack, "Semi-Continuous Hidden Markov Models for Speech Signals," *Computer Speech and Language*, 3: 239–251, 1989.
- [12] F. Jelinek and R.L. Mercer, "Interpolated Estimation of Markov Source Parameters From Sparse Data," *Pattern Recognition in Practice*, E.S. Gelsema and L.N. Kanal, Eds., North-Holland Pub. Co., Amsterdam, pp. 381–397, 1980.
- [13] P.J. Price, W. Fischer, J. Bernstein, and D. Pallett, "A Database for Continuous Speech Recognition in a 1000-Word Domain," *Proc. ICASSP 88*, New York, pp. 651–654, April 1988.
- [14] R. Schwartz et al., "Robust Smoothing Methods for Discrete Hidden Markov Models," *Proc. ICASSP 89*, Glasgow, Scotland, pp. 548–551, May 1989.
- [15] E. Giachin, C.H. Lee, L.R. Rabiner, A.E. Rosenberg, and R. Pieraccini, "On the Use of Interword Context-Dependent Units for Word Juncture Modeling," *Computer Speech and Language*, 6: 197–213, 1992.
- [16] C.H. Lee, E.P. Giachin, and A.E. Rosenberg, "Word Juncture Modeling Using Phonological Rules for HMM Based Continuous Speech Recognition," *CSELT Tech. Reports*, 18 (3): 189–194, June 1990.
- [17] C.H. Lee, L.R. Rabiner, and R. Pieraccini, "Speaker Independent Continuous Speech Recognition Using Continuous Density Hidden Markov Models," *Proc. NATO-ASI, Speech Recognition and Understanding: Recent Advances, Trends and Applications*, P. Laface

- and R. DeMori, Eds., Springer-Verlag, Cetraro, Italy, pp. 135–163, 1992.
- [18] R. Pieraccini and A.E. Rosenberg, "Automatic Generation of Phonetic Units for Continuous Speech Recognition," *Proc. ICASSP 89*, Glasgow, UK, pp. 623–626, May 1989.
 - [19] R. Pieraccini, C.H. Lee, E. Giachin, and L.R. Rabiner, "An Efficient Structure for Continuous Speech Recognition," *Proc. NATO-ASI, Speech Recognition and Understanding: Recent Advances, Trends and Applications*, P. Laface and R. DeMori, Eds., Springer-Verlag, Cetraro, Italy, pp. 211–216, 1992.
 - [20] D. Paul, "A Speaker Stress Resistant Isolated Word Recognizer," *Proc. ICASSP 87*, Dallas, TX, pp. 713–716, April 1987.
 - [21] C.H. Lee, E. Giachin, L.R. Rabiner, R. Pieraccini, and A.E. Rosenberg, "Improved Acoustic Modeling for Large Vocabulary Continuous Speech Recognition," *Computer Speech and Language*, 6: 103–127, 1992.
 - [22] L. Deng et al., "Acoustic Recognition Component of an 86,000 Word Speech Recognizer," *Proc. ICASSP 90*, Albuquerque, NM, pp. 741–744, April 1990.
 - [23] T. Svendsen and F.K. Soong, "On the Automatic Segmentation of Speech Signals," *Proc. ICASSP 87*, Dallas, TX, pp. 77–80, April 1987.
 - [24] L.R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer, and M.A. Picheny, "Automatic Construction of Acoustic Markov Models for Words," *Proc. Int. Symposium on Signal Processing Applications*, (Brisbane, Australia), 565–569, 1987.
 - [25] H.W. Hon and K.F. Lee, "On Vocabulary Independent Speech Modeling," *Proc. ICASSP 90*, Albuquerque, NM, pp. 725–728, April 1990.
 - [26] R. Pieraccini and C.H. Lee, "Factorization of Language Constraints in Speech Recognition," *Proc. 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, CA, June 1991*.