THE ELECTRICAL ENGINEERING AND APPLIED SIGNAL PROCESSING SERIES

Edited by Alexander Poularikas

The Advanced Signal Processing Handbook: Theory and Implementation for Radar, Sonar, and Medical Imaging Real-Time Systems
Stergios Stergiopoulos

The Transform and Data Compression Handbook
K.R. Rao and P.C. Yip

Handbook of Multisensor Data Fusion
David Hall and James Llinas

Handbook of Neural Network Signal Processing
Yu Hen Hu and Jenq-Neng Hwang

Handbook of Antennas in Wireless Communications
Lal Chand Godara

Noise Reduction in Speech Applications
Gillian M. Davis

Signal Processing Noise
Vyacheslav P. Tuzlukov

Digital Signal Processing with Examples in MATLAB®
Samuel Stearns

Applications in Time-Frequency Signal Processing
Antonia Papandreou-Suppappola

The Digital Color Imaging Handbook
Gaurav Sharma

Pattern Recognition in Speech and Language Processing
Wu Chou and Biing Huang Juang

Forthcoming Titles

Propagation Data Handbook for Wireless Communication System Design
Robert Crane

Smart Antennas
Lal Chand Godara

Nonlinear Signal and Image Processing: Theory, Methods, and Applications
Kenneth Barner and Gonzalo R. Arce

Forthcoming Titles *(continued)*

Soft Computing with MATLAB®
Ali Zilouchian

Signal and Image Processing Navigational Systems
Vyacheslav P. Tuzlukov

Wireless Internet: Technologies and Applications
Apostolis K. Salkintzis and Alexander Poularikas

PATTERN RECOGNITION in SPEECH and LANGUAGE PROCESSING

Edited by
WU CHOU
Avaya Labs Research

BIING HWANG JUANG
Georgia Institute of Technology

CRC PRESS

Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Pattern recognition in speech and language processing / edited by Wu Chou and Biing-Hwang Juang.
p. cm.
Includes bibliographical references and index.
ISBN 0-8493-1232-9 (alk. paper)
1. Automatic speech recognition. 2. Pattern recognition systems. I. Chou, Wu. II. Juang, B. H. (Biing-Hwang)

TK7882.S65 P39 2003
006.4′54—dc21 2002191163

Visit the CRC Press Web site at www.crcpress.com

International Standard Book Number 0-8493-1232-9
Library of Congress Card Number 2002191163
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

Preface

Approaches to the problems of designing speech and language processing algorithms for human machine communication used to be taken from the perspectives of linguistics and speech science, until the late 1970s. Due to the advances in computing and statistical modeling, data driven pattern recognition methods have become a fast moving research area during the past two decades and contributed much to the progress in this field. As the era of information age continues to develop, we witness an ever increasing need in intelligent human-machine communications, as well as the creation of machine understandable metadata for Web content and other information sources. This handbook is to fill the need of a systematic and up-to-date presentation of new pattern recognition approaches in speech and language processing.

The book starts with fundamentals and recent theoretical advances in pattern recognition with an emphasis on classifier design criteria and optimization procedures. It covers several recent research advances in this area, such as the minimum error rate (MCE) method, the minimum Bayes risk approach, adaptive system design and decision rules, neural networks, distributed recognizers, and decision fusion. These methods depart from the conventional paradigm which links a classifier design to the classical problem of distribution estimation. Instead, more meaningful criteria are introduced which significantly improve the discrimination power of a classifier, particularly when applied to speech problems in which the notion of data distribution is difficult to realize.

The second part of the book is, therefore, specially focused on the approaches and methods applied to speech processing. It covers topics such as Bayes minimum risk approach to speech recognition, large vocabulary speech recognition based on statistical methods, recognition of spontaneous speech in dialogue interaction, speech and speaker verification, and audio information retrieval and indexing. These chapters provide a comprehensive coverage of recent advances in applying pattern recognition to real systems in speech and audio processing.

The third part of the book is devoted to topics of pattern recognition in language processing. It contains chapters in language modeling based on latent semantic indexing, salient information representation and processing in natural language dialogue system, statistical machine translation, methods in topic detection, tracking, and name identity identification. These topics are new trends in language processing, and significant progress has been made in recent years. It has a direct impact to the practice and implementation of information processing systems for Web content, broadcast news, and other content-rich information resources.

This book is a collective effort, motivated by the excitement of the new advances in

this field and the urgent need to bring these advances to a general audience. The contributing authors of this book are leading experts in the field of speech and language processing. Attempts are made to make each chapter self-contained and comprehensible for readers with general background in pattern recognition and information processing. It is intended to be a handbook or reference textbook for researchers, graduate students, and advanced undergraduate students who want to follow the new advances in pattern recognition. Sufficient references are provided at the end of each chapter to serve as an entry point for an interested reader to pursue further.

We would like to thank all contributors of this book. Without their commitment and quality of work, this book would not be possible. We appreciate the support and encouragement from our colleagues at Avaya Labs Research during the preparation of this book. It was a pleasant working experience with CRC Press - their technical support was very helpful to us.

Wu Chou
Biing-Hwang Juang

Basking Ridge, New Jersey
September, 2002

Contributors

A. Abella
Speech Research
AT&T Laboratories
Florham Park, NJ

James Allan
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA

T. Alonso
Speech Research
AT&T Laboratories
Florham Park, NJ

Jerome R. Bellegarda
Spoken Language Group
Apple Computer, Inc.
Cupertino, CA

William Byrne
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD

Wu Chou
Avaya Labs Research
Basking Ridge, NJ

Sadaoki Furui
Department of Computer Science
Tokyo Institute of Technology
Tokyo, Japan

Jean-Luc Gauvain
LIMSI-CNRS
Université de Paris Sud
Orsay Cedex, France

Vaibhava Goel
T.J. Watson Research Center
IBM
Yorktown Heights, NY

Allen L. Gorin
Speech Research
AT&T Laboratories
Florham Park, NJ

Qiang Huo
Department of Computer Science and Information Systems
The University of Hong Kong
Hong Kong, China

Biing-Hwang Juang
Avaya Labs Research
Basking Ridge, NJ

Shigeru Katagiri
Intelligent Communication Laboratory and Speech Open Laboratory
Nippon Telegraph and Telephone Corporation
Tokyo, Japan

Lori Lamel
LIMSI-CNRS
Université de Paris Sud
Orsay Cedex, France

Qi (Peter) Li
Bell Laboratories

Lucent Technologies
Murray Hill, NJ

John Makhoul
BBN Technologies
Cambridge, MA

Hermann Ney
Lehrstuhl fuer Informatik VI
Human Language Technology and
Pattern Recognition
Computer Science Department
University of Technology
Aachen, Germany

F. J. Och
Lehrstuhl fuer Informatik VI
Human Language Technology and
Pattern Recognition
Computer Science Department
University of Technology
Aachen, Germany

G. Riccardi
Speech Research
AT&T Laboratories
Florham Park, NJ

Richard M. Schwartz
BBN Technologies
Cambridge, MA

J. H. Wright
Speech Research
AT&T Laboratories
Florham Park, NJ

Contents

1

Minimum Classification Error (MCE) Approach in Pattern Recognition

Wu Chou
Avaya Labs Research, Avaya Inc., USA

CONTENTS

1.1 Introduction

Pattern recognition is a fast moving research area. The advent of powerful computing devices and the success of statistical approaches, such as hidden Markov model for speech and language processing, triggered a renewed pursuit for more powerful statistical methods to further reduce the pattern recognition error rate and improve the robustness of the pattern classifier across various adverse conditions. Among this new pursuit, the use of discriminant function methods in pattern recognition has emerged as a promising approach, and it is applied successfully to speech and language processing. This chapter is intended to provide a revisit to the statistical formulation of the minimum classification error (MCE) based discriminative methods in speech and language processing, take a critical view of the approach, provide a comprehensive overview of the field, and hopefully inspire other innovations that would potentially lead to new discriminative methods in pattern recognition.
Although the statistical formulation of MCE based discriminative methods has its root in the classical Bayes decision theory, it departs from the conventional paradigm

This chapter is developed based on "Discriminant-function-based minimum recognition error rate pattern-recognition approach to speech recognition," by Wu Chou, appeared in *Proceedings of The IEEE*, Vol. 88, No. 8, ©2000 IEEE.

which links a recognition task to the problem of distribution estimation. Instead, it takes a discriminant function based statistical pattern classification approach, and for a given family of discriminant function, optimal classifier/recognizer design involves finding a set of parameters which minimize the empirical pattern recognition error rate. The use of discriminant function in pattern recognition was started many years ago. One classical example of using discriminant function for classifier design in statistical literature is the two class classification problem using linear discriminant functions [28, 31]. In particular, a window based method was described in [28] for the two class classification problem using linear discriminant functions that minimize the probability of classification error rate. The focus of this chapter is on the recent development of the general MCE based discriminative methods. The discriminant functions that we encounter are usually non-linear and often related to the structure of the statistical framework used in speech and language processing such as hidden Markov models.

The reason of taking a discriminant function based approach to classifier design, as will be further elaborated, is due mainly to the fact that we lack complete knowledge of the form of the data distribution and training data are inadequate, particularly in dealing with speech and language problems. The performance of a recognizer is normally defined by its expected recognition error rate, and an optimal recognizer should be the one that achieves the least expected rate of recognition error. The difference between the distribution estimation based approach and the discriminant function based MCE approach lies in the way the recognition error is expressed and in the computational steps that would lead to the minimization of such error functions. A key to the development of the MCE method is a new error function which incorporates the recognition operation and performance in a functional form, from which the performance of the classifier can be directly evaluated and optimized.

Classifier design without assuming the knowledge of class posterior probabilities, which are the basis of the distribution estimation based classier design, has been studied in many areas. In particular, Tsypkin [112] and Amari [5, 2] pioneered this approach for self-learning and self-organizing nets. They formulated the problem of self-learning into a classification problem which consists of optimal partitioning of the observation space into regions, X_k, for which the expected risk, R, is minimized. In addition, a mathematical minimization procedure, generalized probabilistic descent (GPD) algorithm or stochastic approximation, was proposed as a means for classifier design under this framework. Since then, various loss functions have been used in designing classifiers, including those popular mean-square error based loss functions. However, many tractable loss functions do not have a direct relation to the recognition error rate minimization, and therefore, albeit based on discriminant functions, they are not directly related to recognition error rate which should be the most sensible choice for classifier design.

Over the past decade, the MCE based approach has been developed to overcome the fundamental limitations of the traditional approach and to directly link the classifier design problem to classification error rate minimization. In order to alleviate the dependency on the class posterior distributions, a discriminant function based MCE approach was proposed by Juang et al. [50] as an alternative to optimal classifier de-

sign. Although this approach applies to the pattern recognition problem in general, it finds various applications in speech and language processing. It was first applied to dynamic time warping based pattern recognition systems [16, 56]. Application to hidden Markov model based continuous speech recognition systems was formulated as a segmental and string model based MCE approach [18, 19], and successful applications of this approach were reported in [20, 27, 35, 39, 80, 81]. This approach was further extended to form a combined string model, in which training of other model components in speech and language processing can be achieved under a unified MCE framework [22, 41]. It was applied to discriminative model combination [13, 79] and to applications in speaker identification and verification [74, 36, 62]. The basic idea of the MCE approach was further developed for applications in utterance verification problems [101, 111, 77]. A general framework of combining detection and verification in speech recognition and understanding was also proposed, in which the discriminant function based pattern recognition approach was applied in both detection and verification processes [54, 60].

We begin in the next section with a brief review of the Bayes decision theory and its application to the formulation of statistical pattern recognition problem. We introduce the discriminant function based statistical pattern recognition approach in Section 3. In Section 4, we provide a brief introduction to speech recognition and hidden Markov modeling. The discriminant function based MCE pattern recognition approach and its application to HMM based speech recognition systems are introduced in Section 5. Comparisons are made to other criteria in speech recognition and in particular, we study the relation between MCE and MMI (maximum mutual information) criteria in classifier design in the second half of Section 5. In Section 6, we study the embedded string model based MCE approach and its extension to the higher level combined string model. We discuss issues and applications in discriminative model combination, discriminative language model estimation, and discriminative feature extraction under the general theoretical framework of the combined string model. Section 7 is devoted to applications of discriminant function based pattern recognition approach in verification and identification. The discriminant function approach is studied for various applications in speech and language processing, such as speaker identification and verification, utterance verification, recognition based on generalized confidence measures, detection and verification based approach in speech recognition and understanding. The chapter is summarized with discussions in Section 8.

1.2 Optimal Classifier from Bayes Decision Theory

For an M class classification problem, a classifier is to classify each random sample x into one of the M classes. We denote these classes by C_i, $i = 1, 2, \ldots, M$. The classifier $C(x)$ defines a mapping from the sample space $x \in X$ to the discrete

categorical set $C_i \in Y$. Let $P(x, C_i)$ be the joint probability distribution of x and C_i, a quantity which is assumed to be known to the designer of the classifier. In other words, the designer has full knowledge of the random nature of the source. From the set of joint probability distributions, the marginal and the conditional probability distributions can be easily calculated.

In order to characterize the performance of the classifier, every class pair (j, i) can be associated with a cost or loss function e_{ji} which signifies the cost of classifying (or recognizing) a class i observation into a class j event. The loss function is generally non-negative with $e_{ii} = 0$ representing correct classification. The loss function is a function from $X \times Y \rightarrow R$ where R is the set of real numbers. In classification, we make a decision $C(x)$ for observing a random sample x. Since $P(C_j \mid x)$ is the class posterior probability that the random input x is from C_j, the average loss associated with making a decision $C(x) = C_i$ can be defined as [31]

$$R(C_i \mid x) = \sum_{j=1}^{M} e_{ji} P(C_j \mid x). \tag{1.1}$$

This leads to a reasonable performance measure for the classifier, i.e., the expected loss, defined as

$$\mathcal{L} = \int R(C(x) \mid x) dP(x) \tag{1.2}$$

where $C(x)$ represents the classifier's decision (assuming one of the M "values," C_1, $C_2, \ldots, C_M$), based on a random observation x drawn from a probability distribution $P(x)$. The decision function, $C(x)$, depends on the classifier design. Obviously, if the classifier is so designed that for every x

$$R(C(x) \mid x) = \min_i \; R(C_i \mid x) \; , \tag{1.3}$$

the expected loss in equation (1.2) will be minimized.

For many applications, including speech recognition, the loss function e_{ij} is usually chosen to be the zero-one loss function defined by

$$e_{ij} = \begin{cases} 0, \; i = j \\ 1, \; i \neq j \end{cases} \quad i, j = 1, 2, \ldots, M \tag{1.4}$$

which assigns no loss to correct classification and a unit loss to any error, regardless of the class. With this type of loss function, the expected loss $\mathcal{L}$ is thus the error probability of classification or recognition. The conditional loss becomes

$$R(C_i \mid x) = \sum_{i \neq j} P(C_j \mid x) = 1 - P(C_i \mid X). \tag{1.5}$$

The optimal classifier that achieves minimum $\mathcal{L}$ is thus the one that implements the following:

$$C(x) = C_i \quad \text{if} \quad P(C_i \mid x) = \max_j P(C_j \mid x). \tag{1.6}$$

For minimum error rate classification, the classifier employs the decision rule of (1.6) which is called the "maximum a posterior" (MAP) decision. The minimum error rate achieved by MAP decision is called "Bayes risk". When all posterior probabilities are known, the classifier based on MAP rule is an optimal classifier based on the Bayes decision theory. However, if these probabilities are not known or the decision rule is not based on the class posterior probability, then we cannot use this result directly.

In practice, these probabilities have to be estimated from a training data set with known class labels. The classical Bayes decision theory thus effectively transforms the classifier design problem into a distribution estimation problem. This is the basis of the Bayesian statistical approach to pattern recognition which can be stated as: given (or collect) a set of training data (observations) $\{x_1, x_2, \ldots, x_K\}$ with known class labels, estimate the a posterior probabilities $P(C_i \mid x)$, $i = 1, 2, \ldots, M$ for any x to implement the maximum a posterior decision for minimum Bayes risk. The a posterior probability $P(C_i \mid x)$ can be rewritten as

$$P(C_i \mid x) = P(x \mid C_i)P(C_i)/P(x) \; . \tag{1.7}$$

Since $P(x)$ is not a function of the class index and thus has no effect in the MAP decision, the needed probabilistic knowledge can be represented by the class prior $P(C_i)$ and the conditional probability $P(x \mid C_i)$.

There are several issues associated with this classical approach. First, the distributions usually have to be parameterized in order for them to be practically useful for the implementation of the MAP rule of (1.6). The classifier designer therefore has to determine the right parametric form of the distributions. For most of the real world problems, this is a difficult task. Our choice of the distribution form is often limited by the mathematical tractability of the particular distribution functions and is very likely to be inconsistent with the actual distribution. This means that the true MAP decision can rarely be implemented and the minimum Bayes risk generally remains an unachievable lower bound. Second, given a parameterized distribution form, the unknown parameters defining the distribution have to be estimated from a finite amount of labeled training data, requiring that the estimation method has to be able to produce consistent parameter values when the size of the training samples varies. Third, it requires a training data set of sufficient size in order to have reliable parameter estimates. But in practice and for speech and language processing in particular, training data are always sparse compared to all possible realizations and variations in human speech and language. These three basic issues point out a fundamental fact; that is, despite the conceptual optimality of the Bayes decision theory and its applications to pattern recognition, it cannot always be accomplished in practice, because most practical "MAP" decisions in speech and language processing are not true MAP decisions. This understanding is critical for the discussion that follows.

1.3 Discriminant Function Approach to Classifier Design

Discriminant functions on the other hand are those functions which characterize the decision rule of the classifier. They may or may not be probability or likelihood based functions, and they can come from different parametric families, including those families which have no relation to the parametric form of the class posterior distribution $P(C_i \mid x)$ as required in the classical Bayes decision theory. One well studied family of discriminant function is the linear discriminant function which has computational advantages and due to its analytic form, has received considerable attention and theoretical development for its design. To illustrate the concept, we consider the case of a two class $\{M_1, M_2\}$ classification problem. The classifier uses a discriminant function $g(x)$ such that

$$\begin{cases} \text{if} \quad g(x) \geq 0, \text{ then x is classified to } M_1 \\ \text{if} \quad g(x) < 0, \text{ then x is classified to } M_2 \end{cases} \tag{1.8}$$

Linear discriminant functions are those functions of the form

$$g(x) = w^T x + w_0 \tag{1.9}$$

where $w^T = [w_1, w_2, \cdots, w_k]$ and w_0 a real number. Or more generally,

$$g(x) = w_0 + w_1\phi_1(x) +, \ldots, +w_k\phi_k(x) = a^T y(x) \tag{1.10}$$

where

$$\begin{aligned} a^T &= [w_0, w_1, \ldots, w_k] = [w_0, w^T] \\ y^T &= [1, \phi_1, \ldots, \phi_k] = [1, \Phi^T] \end{aligned} \tag{1.11}$$

where T is the transposition notation, and the ϕ_i are known linearly independent functions of x.

For M class classification problem using discriminant functions, a set of discriminant functions $\{g_i(x) \mid i = 1, \ldots, M\}$ are used, and the classifier $C(x)$ is defined such that

$$C(x) = I \qquad \text{iff} \qquad I = \underset{i}{\operatorname{argmax}}\, g_i(x). \tag{1.12}$$

When the loss function $R(C(x) \mid x)$ is specified, the problem of optimal classifier design using discriminant functions becomes a minimization problem of finding a best set of discriminant functions $\{g_i(x) \mid i = 1, \ldots, M\}$ from a class of discriminant functions which minimizes the expected loss $\mathcal{L}$ as defined in Eq. (1.2). In other words, the classifier design problem is to find

$$\{\bar{g}_i(x) \mid i = 1, \ldots, M\} = \underset{g_i(x) \in \mathcal{F}(X \times Y)}{\operatorname{argmin}} \int R(C(x) \mid x) dP(x) \tag{1.13}$$

where $\mathcal{F}(X \times Y)$ is the given family of discriminant functions. If the loss function is given as in Eq. (1.4) and the particular set of discriminant functions used in the classifier are the "true" class posterior probability $P(C_i \mid x)$, then Eq. (1.12) implements the same MAP decision rule as defined in Eq. (1.6). However, it is important to point out that the discriminant function approach to the optimal classifier design as specified in Eq. (1.13) often has an infinite number of solutions even for the same classifier. It is easy to see that if $\{\bar{g}_i(x) \mid i = 1, \ldots, M\}$ is an optimal solution to Eq. (1.13), then for any $\{(a, b) \mid a > 0, b \in R\}$, $\{a\bar{g}_i(x) + b \mid i = 1, \ldots, M\}$ is another optimal solution and defines the same classifier. Again, this is quite different from the distribution estimation based approach in pattern classification. If the discriminant functions are limited to the class posterior probabilities, any deviation from the "true" class posterior probability $P(C_i \mid x)$ will result in a different classifier and will be inferior to the optimal MAP classifier that achieves the minimum Bayes risk.

The use of discriminant functions in statistical pattern recognition is to solve the classifier design problem when the exact form and value of the class posterior probabilities $P(C_i \mid x)$ are not known (even with the help of training data) or the classifier has to be based on a particular class of discriminant functions. These discriminant functions in the classifier may come from either the model used to characterize the generation process of the recognition objects or the practical consideration of mathematical tractability and algorithmic complexity. Classifying human speech meets both scenarios. In particular, the method of hidden Markov modeling is a prevalent approach in providing statistical characterization of human speech, and the full complexity of classifying spontaneous human speech is still too great to handle. In the next section, we give a brief discussion of the speech recognition problem and HMM-based acoustic modeling before introducing a discriminant function based approach to speech recognition.

1.4 Speech Recognition and Hidden Markov Modeling

Speech recognition is a problem of recognizing a word sequence from human speech. It can be viewed as a communication problem. The human brain serves as the text generator which generates the word string $\mathcal{W}$. The word string goes to the acoustic channel which consists of a speaker's articulatory apparatus and other acoustic processes that convert the text string into an audible acoustic waveform. The acoustic channel in verbal communication acts as a data transducer and composer. The speech recognizer is a decoder which performs an inverse operation to decode the message from the speech waveform. Therefore, a decoder performs a maximum a posterior decision that determines the word sequence $\hat{\mathcal{W}}$ such that

$$\hat{\mathcal{W}} = \underset{W}{\operatorname{argmax}}\, P(W \mid X) = \underset{W}{\operatorname{argmax}}\, P(X \mid W)P(W), \tag{1.14}$$

where $P(X \mid W)$ is the score from acoustic modeling, and $P(W)$ is the score from the language model. A typical speech recognition system consists of the following basic components.

- Acoustic feature extraction: Acoustic feature extraction is to extract the features for speech recognition from the speech waveform. It typically includes a short-time cepstral analysis which generates a feature vector of low frequency (10-16) cepstral coefficients for every $10ms$. Various signal processing procedures are performed to separate the salient acoustic information for speech recognition purposes. From now on, we will use the notation $X = (x_1, \ldots, x_T)$ to represent the acoustic observation feature vector sequence.

- Acoustic modeling: Acoustic modeling provides statistical modeling for the acoustic observation X. Hidden Markov modeling is the prevalent choice for this purpose, although the neural network based approach is also used in many systems. The model units can be based on semantically meaningful units, such as words, or phonetically meaningful subword units such as phonemes.

- Language modeling: Language modeling provides linguistic and grammar constraints to the text sequence W. It is often based on statistical N-grams language models. An N-gram language model is of the form $P(w_n \mid w_1, \ldots, w_{n-1})$, which is the probability of observing word w_n given the word history $w_1, \ldots, w_{n-1}$.

- Decoding engine: The decoding engine searches for the best word sequence given the feature and the model. For speech recognition based on HMM modeling, this is achieved through Viterbi decoding. For a discrete observation probability based system, word string $\mathcal{W}$ is given by

$$\hat{\mathcal{W}} = \underset{W}{\operatorname{argmax}} P(X, \Lambda_{W_Q} \mid W), \tag{1.15}$$

  where Λ_{M_Q} is the best state sequence given W, X and the model Λ. For continuous density HMMs,

$$\hat{\mathcal{W}} = \underset{W}{\operatorname{argmax}} \log f(X, \Lambda_{W_Q} \mid W), \tag{1.16}$$

  which is based on the log-likelihood score along the best state sequence W_Q.

1.4.1 Hidden Markov Modeling of Speech

Speech is generated from human articulator, and it is unique in many ways. When we speak, our articulatory apparatus (the lips, jaw, tongue, and velum) modulates the air pressure and flow to produce an audible sequence of sounds. Due to the physical constraints, the articulator configuration cannot undergo very drastic changes, and during the short interval where the articulatory configuration stays relatively constant, a region of "quasi-stationarity" in the produced speech can often be observed.

Hidden Markov modeling is a powerful statistical framework for time varying quasi-stationary process and a popular choice for statistical modeling of speech signal.
Given a speech utterance, let $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ be a feature vector sequence extracted from the speech waveform, where $\mathbf{x}_t$ denotes a short-time vector measurement, and it is conventionally a cepstral vector.
Further consider a first-order N-state Markov chain governed by a state transition probability matrix $\mathcal{A} = [a_{ij}]$, where a_{ij} is the probability of making a transition from state i to state j. Assume that at $t = 0$ the state of the system q_0 is specified by an initial state probability $\pi_i = P(q_0 = i)$. Then, for any state sequence $\mathbf{q} = (q_0, q_1, \ldots, q_T)$, the probability of $\mathbf{q}$ being generated by the Markov chain is

$$P(\mathbf{q} \mid \mathcal{A}, \pi) = \pi_{q_0} a_{q_0 q_1} a_{q_1 q_2} \cdots a_{q_{T-1} q_T} \ . \tag{1.17}$$

Suppose the system, when at state q_t, puts out an observation $\mathbf{x}_t$ according to a distribution $b_{q_t}(\mathbf{x}_t) = P(\mathbf{x}_t \mid q_t), q_t = 1, 2, \ldots, N$. The hidden Markov model used as a distribution for the speech utterance X is then defined as

$$\begin{aligned} P\left(X \mid \pi, \mathcal{A}, \{b_j\}_{j=1}^N\right) &= P(X \mid \Lambda) = \textstyle\sum_{\mathbf{q}} P(X, \mathbf{q} \mid \Lambda) \\ &= \textstyle\sum_{\mathbf{q}} P(X \mid \mathbf{q}, \Lambda)\, P(\mathbf{q} \mid \Lambda) = \sum_{\mathbf{q}} \pi_{q_0} \prod_{t=1}^T a_{q_{t-1} q_t} b_{q_t}(\mathbf{x}_t) \end{aligned} \tag{1.18}$$

where $\Lambda = (\pi, \mathcal{A}, \{b_j\}_{j=1}^N)$ is the parameter set for the model.
As can be seen in (1.18), $\{b_{q_t}\}$ defines the distribution for short-time observations and $\mathcal{A}$ characterizes the behavior and interrelationship between different states of the speech generation process. In other words, the structure of a hidden Markov model provides a reasonable means for characterizing the distribution of a speech signal. Normally, N, the total number of states, is much smaller than T, the time duration of the speech utterance. The state sequence $\mathbf{q}$ displays a certain degree of stability among adjacent q_t's due to the above mentioned "quasi-stationarity". The use of HMMs as speech distributions is shown to be practically effective.
It should be noted that the choice of state observation distributions $b_{q_t}(\mathbf{x}_t)$ is not specified. Different choices of speech dimensions for the observation space may require different forms of the state observation distribution. For cepstral vectors, a mixture Gaussian density is commonly employed. Moreover, regardless of the practical effectiveness of HMM in speech recognition, it should not be taken as the true distribution form of speech, and therefore any recognition system or decision rule that operates based on HMM is not going to achieve the minimum error rate as implied in the true Bayes MAP decision.
In order to apply HMMs to speech recognition, three basic problems have to be resolved, namely, the evaluation problem, the decoding problem, and the estimation problem [89, 47]. The evaluation problem is to estimate the probability $P(X \mid \Lambda)$ of observing the speech feature vector sequence X given the hidden Markov model. The decoding problem is to find a best state sequence $\mathbf{q}$ which is optimal in a certain sense given the speech feature sequence X. Since states in HMM are related to words and word classes, the word sequence in speech utterance can be identified by tracing through the word labels in state sequence $\mathbf{q}$. The estimation problem is to estimate

HMM parameters Λ from a given set of training samples according to some meaningful criterion. The conventional approach is based on the maximum likelihood (ML) principle, and the model parameter set Λ is estimated so that the likelihood on the training data is maximized. Various highly efficient ML based algorithms are developed in speech recognition for HMMs, such as Baum-Welch algorithm [7] and segmental k-means algorithm [49]. More discussions of parameter estimation problem for HMMs can be found in [47]. It should be noted that the conventional ML method in speech recognition does not necessarily lead to a minimum error rate performance for the recognizer. This is due mainly to 1) the likely mismatch between the chosen distribution form and the actual speech data distribution, and 2) the finite training (known) data set which is often inadequate.

1.5 MCE Classifier Design Using Discriminant Functions

As it is noted, without the knowledge of the form of the class posterior probabilities required in the classical Bayes decision theory, classifier design by distribution estimation often does not lead to an optimal performance. This motivates the effort of searching for other alternative criteria in classifier design. In particular, criteria of MMI (maximum mutual information) and MDI (minimum discriminative information) are used in many applications [4, 84]. Although these methods demonstrate significant performance advantages over the traditional ML approach, they are not based on a direct minimization of a loss function which links to the classification error rate.

Do-Tu et al. [28] studied MCE solution for the two class non-parametric classification problem using linear discriminant functions. They employed a windowed scheme to overcome the problem of singular gradient functions associated with the error count indicator function of the classifier. A general approach for multi-class and non-linear discriminant functions are proposed by Juang et al. [50]. This general approach is called "minimum classification error (MCE) method" in which the classifier design and parameter estimation are to correctly discriminate the observations for best recognition/classification results rather than to fit the distributions to the data.

1.5.1 MCE Classifier Design Strategy

Let us consider a set of class discriminant functions $g_i(x;\Lambda)$, $i = 1, 2, \ldots, M$, defined by the parameter set Λ. The classifier $C(x)$ is the one that for an object x:

$$C(x) = I \qquad \text{iff} \qquad I = \underset{i}{\operatorname{argmax}}\, g_i(x). \tag{1.19}$$

The general MCE classifier design strategy is based on a special type of loss function. Parameters of the classifier are estimated in such a way that minimizing the expected

loss relates to a minimization of the recognition error rate of the classifier. This is achieved through a three step process.
1) The misclassification measure in the MCE based approach is defined as

$$d_i(X) = -g_i(X;\Lambda) + \log\left[\frac{1}{M-1}\sum_{j,j\neq i}\exp[\mathrm{g_j(X;\Lambda)}\eta]\right]^{1/\eta} \tag{1.20}$$

where η is a positive number [18]. This misclassification measure is a continuous function of the classifier parameters Λ and attempts to emulate the decision rule given the discriminant function $g_i(x)$. For an i^{th} class utterance X, $d_i(X) > 0$ implies misclassification and $d_i(X) \leq 0$ means a correct decision. When η approaches ∞, the term in the bracket is the L^η norm on the discrete integer set $\{j \mid j \neq i, j = 1,\ldots,M\}$ which converges to the $\|\ \|_\infty$ norm and becomes $\max_{j,j\neq i} g_j(X;\Lambda)$. By varying the value of η and M, one can take all the competing classes into consideration, according to the individual significance, when searching for the classifier parameter Λ.
2) The loss function is used for recognition error rate minimization. The misclassification measure of (1.20) is embedded in a smooth zero-one function, for which any member of the sigmoid function family is an obvious candidate. A general form of the *loss function* can then be defined as:

$$\ell_i(X;\Lambda) = \ell(d_i(X)) \tag{1.21}$$

where ℓ is a sigmoid function, one example of which is

$$\ell(d) = \frac{1}{1+\exp(-\gamma d+\theta)} \tag{1.22}$$

with θ normally set to 0 and γ set to greater or equal to one. Clearly, when $d_i(X)$ is much smaller than zero, which implies correct classification, virtually no loss is incurred. When $d_i(X)$ is positive, it leads to a penalty which becomes essentially a classification/recognition error count.
3) The classifier parameter estimation is based on the minimization of the expected loss. For any unknown object X, the classifier performance is measured by

$$\ell(X;\Lambda) = \sum_{i=1}^{M}\ell_i(X;\Lambda)1(X\in C_i) \tag{1.23}$$

where $1(\cdot)$ is the indicator function. The expected loss, which is related to recognition error rate, is given by

$$L(\Lambda) = E_X[l(X;\Lambda)]. \tag{1.24}$$

This three-step definition emulates the classification operation as well as the recognition error rate based performance evaluation in a smooth functional form, suitable

for classifier parameter optimization. It should be pointed out that if the correct form of the posterior probability $P_\Lambda(C_i \mid x)$ is used, the Bayes minimum risk is then expressed as

$$L(\Lambda) = \sum_{k=1}^{M} \int_{\chi_k} P_\Lambda(C_k \mid x) 1(x \in C_k) dP(x), \tag{1.25}$$

where $\chi_k = \{x \in \chi \mid P_\Lambda(C_k \mid x) \neq \max_j P_\Lambda(C_j \mid x)\}$, and χ represents the entire signal space. This can be approximated by the loss function in MCE approach as follows:

$$\begin{aligned} L(\Lambda) &= \sum_{k=1}^{M} \int_{\chi_k} P_\Lambda(C_k \mid x) 1(x \in C_k) 1(P_\Lambda(C_k \mid x) \neq \max_j P_\Lambda(C_j \mid x)) dP(x) \\ &\simeq \sum_{k=1}^{M} \int_{\chi_k} P_\Lambda(C_k \mid x) 1(x \in C_k) \ell_k(d_k(x;\Lambda) dP(x). \end{aligned} \tag{1.26}$$

An important point here is that approximation accuracy of Eq. (1.26) can be controlled by varying the constants in the smooth MCE loss function. Based on the criterion of (1.23), we can choose to minimize one of two quantities for the classifier parameter search; one is the expected loss and the other the empirical loss.

1.5.2 Optimization Methods

The purpose of the training process in the MCE approach is to find a set of parameters Λ so that a prescribed loss is minimized. As mentioned previously, the two kinds of loss we focus on are the expected loss and the empirical loss.

1.5.2.1 Expected Loss

For a classification problem involving M different classes, the expected loss is defined as

$$L(\Lambda) = E_X\{\ell(X;\Lambda)\} = \sum_{i=1}^{M} \int_{X \in C_i} \ell_i(X;\Lambda) dP(X). \tag{1.27}$$

Various minimization algorithms can be used to minimize the expected loss. The generalized probabilistic descent (GPD) algorithm is a powerful algorithm that can be used to accomplish this task [2]. In GPD based minimization algorithm, the target function $L(\Lambda)$ is minimized according to an iterative procedure:

$$\Lambda_{t+1} = \Lambda_t - \epsilon_t U_t \nabla \ell(X_t, \Lambda) \mid_{\Lambda=\Lambda_t} \tag{1.28}$$

where U_t is a positive definite matrix [25], ϵ_t is a sequence of positive numbers, and $\nabla \ell(X_t, \Lambda) \mid_{\Lambda=\Lambda_t}$ is the gradient function of the loss function at $\Lambda = \Lambda_t$, and X_t is the t-th training sample used in the sequential training process.
The convergence properties of GPD algorithm was studied in the literature (e.g. [25, 15, 100, 30]) and sometimes under the name of stochastic approximation. Under very general conditions, the following convergence properties can be established [25]

Property 1 *Suppose the following conditions are satisfied:*

$$C1 \;:\quad \sum_{t=1}^{\infty} \epsilon_t = \infty, \quad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty \quad \epsilon_t \geq 0;$$

$$C2 \;:\quad \exists\, 0 \leq V < \infty, \quad \textit{such that for all t, the inner product}$$

$$R_t(\epsilon_t, \theta_t) =< \nabla \ell(X, \Lambda_t), H(X, \Lambda_t + \epsilon_t \theta_t \nabla \ell(X, \Lambda_t)) \nabla \ell(X, \Lambda_t) >\leq V,$$

where H *is the Hessian matrix of second order partial derivatives;*

$$C3 \;:\quad \Lambda^* = \arg\min_{\Lambda} E_X\{\ell(X, \Lambda)\} \quad \textit{is the unique } \Lambda \textit{ such that}$$

$$\nabla L(\Lambda) \mid_{\Lambda=\Lambda^*} = \nabla E_X\{\ell(X, \Lambda)\} \mid_{\Lambda=\Lambda^*} = 0.$$

Then, Λ_t *given by*

$$\Lambda_{t+1} = \Lambda_t - \epsilon_t \nabla \ell(X_t, \Lambda) \mid_{\Lambda=\Lambda_t} \tag{1.29}$$

will converge to Λ^* *almost surely (i.e. with probability one).*

Condition $C3$ can be considerably weakened. Even without condition $C3$ the following is still true:

$$E_X \nabla \ell(X, \Lambda_{t_k}) \to 0 \tag{1.30}$$

where Λ_{t_k} is a subsequence of Λ_t. In this case, Λ_{t_k} will converge to a local minimum point Λ^* where $\nabla L(\Lambda) \mid_{\Lambda=\Lambda^*} = 0$.

Adapting the model parameters using a sample by sample updating formula as in Eq. (1.28) is most efficient in terms of the use of available training samples. But the single sample based gradient estimation can be noisy, leading to fluctuations during the parameter estimation process. Batch mode adaptation schemes based on a gradient estimate which is an average of every K samples can also be used. Other variations to the original GPD algorithms are also possible choices to speed up the convergence and reduce the fluctuation during the classifier training process. The convergence properties of these related adaptation algorithms are based on various statistical convergence theories such as martingale theory, potential functions, etc., and it is still a very active area of research. However, from an application point of view, in order to apply this algorithm to speech recognition, such as a speech recognition system using HMMs, the GPD algorithm has to accommodate various constraints imposed on the HMM structures. In particular, the GPD algorithm is an unconstrained minimization scheme that needs modification for solving minimization problems with constraints. As will be shown shortly, one can utilize parameter space transformations to resolve this issue. In this method, the original parameters are updated through the inverse transform from the transformed parameter space to the original parameter space. This is done in such a way that constraints on the original parameters are always maintained. More detailed illustrations of this approach are given in later sections.

1.5.2.2 Empirical Loss

For a given training data set consisting of I samples $\{X_1, \cdots, X_I\}$, the empirical probability measure P_I defined on the training data set is a discrete probability mea-

sure which assigns equal mass at each sample. The empirical loss, on the other hand, is thus expressed as

$$L_0(\Lambda) = \frac{1}{I}\sum_{j=1}^{I}\sum_{i=1}^{M}\ell_i(X_j;\Lambda)1(X_j \in C_i) = \int \ell(X;\Lambda)dP_I \tag{1.31}$$

where j denotes the index of the training utterance X_j in the training set of size I, and P_I is the empirical measure defined on the training set. If the training samples are obtained by an independent sampling from a space with a fixed probability distribution P, the empirical probability distribution P_I will converge to P in distribution as $I \to \infty$. In other words, for any measurable function f,

$$\int f dP_I \to \int f dP. \tag{1.32}$$

The empirical loss defined on the I independent training samples will converge to the expected loss, as the sample size I increases. With sufficient training samples, the empirical loss is an estimate of the expected loss. The goodness of this estimate is determined by the training sample size I, and the convergence rate of the empirical probability measure P_I to the limit distribution P. Various upper bounds on the convergence rate of the empirical probability measure can be found in [87].

1.5.3 Other Optimization Methods

It should be pointed out that although the GPD type of adaptation algorithm is effective and most popular, other optimization methods can also be used for error rate minimization. MCE based classifier design is very specific on the form and structure of the discriminant function and loss function regarding the classifier, and relatively unrestricted to what particular optimization methods which are used to minimize the loss. Many innovations are possible for better optimization results. In particular, methods of linear programming [85], gradient projection [40], and growth-transformation [37, 52, 84] are also used for minimization of the expected loss in MCE classifier design. In the growth transformation based approach, the goal is to seek a transformation $T(x)$ such that $f(Tx) \geq f(x)$, where x is a probability vector (i.e., $\sum_{i=1}^{m} x_i = 1$ and $x_i \geq 0$). This approach, sometimes referred to as extended Baum-Welch (BW) algorithm in some literature, was originated from Baum-Eagon's inequality for $f(x)$ being a polynomial with non-negative coefficients and homogeneous of degree d in its variables. It is extended to rational functions and applied to speech recognition for maximum mutual information (MMI) training with discrete probability codebooks [37]. Later it was further generalized to analytic functions [52]. Since then, this approach was adopted for MCE training of HMM based speech recognition systems with discrete probability densities [94].

In the growth-transformation based optimization approach, the model parameter θ which is a component of the probability vector Θ is updated with the following re-

estimation like formula

$$\hat{\theta} = \frac{\theta \cdot (\frac{\partial f(\Theta)}{\partial \theta} + D)}{\sum_{\theta'} \theta' \cdot (\frac{\partial f(\Theta)}{\partial \theta'} + D)} \tag{1.33}$$

where D is a constant to be determined, and the sum in the denominator is taken over all parameters belonging to the same distribution Θ. It is shown in [37] that there is a value $D(\Theta)$ such that if $D \geq D(\Theta)$, Eq. (1.33) is a growth-transformation and $f(T(\Theta)) = f(\hat{\Theta}) \geq f(\Theta)$. However, Eq. (1.33) cannot be directly applied to parameter estimation of continuous probability densities and new formulations for continuous density HMMs are needed. Using a discrete approximation argument, the growth-transformation method is extended to MMI-based parameter estimation with continuous density HMMs, and new formulations of the growth-transformation for continuous density HMMs in MMI training are derived [84]. More recently, an elegant proof of the theoretical properties of the growth-transformation method for MMI training is given in [3] which establishes the growth-transformation method in a more general setting. But more work remains to be done in order to apply the similar method for general MCE based parameter estimation with continuous probability densities. It is interesting to note that if D is large then the convergence of this algorithm is slow and if D is too large, the algorithm is practically not useful. In order to get fast convergence, D needs to be as small as possible. Modifications of the original algorithm and using search heuristics are attempted to speed up the convergence, and performance improvements over the original approach are also reported in [37, 84].

1.5.4 HMM as a Discriminant Function

Following (1.18), we have several ways of using an HMM as the discriminant function. A basic component in (1.18) is the joint observation-state probability

$$g_i(X, \mathbf{q}; \Lambda) = P^{(i)}(X, \mathbf{q}; \Lambda) = \pi^{(i)}_{q_0} \prod_{t=1}^{T} a^{(i)}_{q_{t-1} q_t} b^{(i)}_{q_t}(x_t), \tag{1.34}$$

which is now defined as a component function $g_i(X, \mathbf{q}; \Lambda)$ for class i as well. The discriminant function for class i can take several possible forms based on $g_i(X, \mathbf{q}; \Lambda)$:

1) $$g_i(X; \Lambda) = \sum_{\mathbf{q}} g_i(X, \mathbf{q}; \Lambda), \tag{1.35}$$

2) $$g_i(X; \Lambda) = \max_{\mathbf{q}} g_i(X, \mathbf{q}; \Lambda), \tag{1.36}$$

3) $$g_i(X; \Lambda) = \left\{ \frac{1}{Q} \sum_{\mathbf{q}}^{Q} g_i(X, \mathbf{q}; \Lambda)^{\alpha} \right\}^{1/\alpha} \tag{1.37}$$

where Q is the total number of possible state sequences, and α is a positive number, and

4) Functions of the above.

Note that (1.35) is equivalent to the likelihood function, (1.36) is equivalent to the maximum joint observation-state sequence probability, and (1.37) is a generalized mixture model which approaches (1.36) when $\alpha \to \infty$. We use the logarithm of (1.36) as an example in our derivation because it is the most popular choice for HMM based recognition systems associated with Viterbi decoding.
The algorithm based on (1.36) is often called *segmental GPD* [18].
We define, for $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ and $\mathbf{x}_t = [x_{t1}, x_{t2}, \ldots, x_{tD}]'$ with D being the dimension of $\mathbf{x}_t$,

$$\begin{aligned} g_i(X;\Lambda) &= \log\left\{\max_{\mathbf{q}} g_i(X,\mathbf{q};\Lambda)\right\} = \log\{g_i(X,\overline{\mathbf{q}};\Lambda)\} \\ &= \sum_{t=1}^{T}\left[\log a^{(i)}_{\overline{q}_{t-1}\overline{q}_t} + \log b^{(i)}_{\overline{q}_t}(\mathbf{x}_t)\right] + \log \pi^{(i)}_{\overline{q}_0} \end{aligned} \tag{1.38}$$

where $\overline{\mathbf{q}} = (\overline{q}_0, \overline{q}_1, \ldots, \overline{q}_T)$ is the optimal state sequence that achieves $\max_{\mathbf{q}} g_i(X,\mathbf{q};\Lambda)$.
We also assume that

$$b^{(i)}_j(\mathbf{x}_t) = \sum_{k=1}^{K} c^{(i)}_{jk}\mathcal{N}\left[\mathbf{x}_t;\mu^{(i)}_{jk},R^{(i)}_{jk}\right] \tag{1.39}$$

where $\mathcal{N}[\ \cdot\ ]$ denotes a normal distribution, $c^{(i)}_{jk}$ are the mixture weights, $\mu^{(i)}_{jk} = [\mu_{jk\ell}]^D_{\ell=1}$ the mean vector, and $R^{(i)}_{jk}$ the covariance matrix which, for simplicity, is assumed to be diagonal, i.e., $R^{(i)}_{jk} = [\sigma^{2^{(i)}}_{jk\ell}]^D_{\ell=1}$.
It may be desirable to maintain the original constraints in the HMM as probability measure, such as: 1) the function being non-negative, 2) $\sum_j a_{ij} = 1$ for all i, and 3) $\sum_k c^{(i)}_{jk} = 1$ for all j and etc. Also, we assume $\sigma^{(i)}_{jk\ell} > 0$. The following parameter transformations allow us to maintain these constraints during parameter adaptation:

$$1) \qquad a_{ij} \longrightarrow \widetilde{a}_{ij} \qquad \text{where } a_{ij} = \frac{e^{a^{\sim}_{ij}}}{\sum_k e^{a^{\sim}_{ik}}}\ , \tag{1.40}$$

$$2) \quad c^{(i)}_{jk} \longrightarrow \widetilde{c}^{(i)}_{jk} \qquad \text{where } c^{(i)}_{jk} = \frac{exp(\tilde{c_{jk}}^{(i)})}{\sum_k exp(\tilde{c_{ik}}^{(i)})}, \tag{1.41}$$

$$3) \qquad \mu^{(i)}_{jk\ell} \longrightarrow \widetilde{\mu}^{(i)}_{jk\ell} = \frac{\mu^{(i)}_{jk\ell}}{\sigma^{(i)}_{jk\ell}}, \quad \text{and} \tag{1.42}$$

$$4) \qquad \sigma^{(i)}_{jk\ell} \longrightarrow \widetilde{\sigma}^{(i)}_{jk\ell} = \log \sigma^{(i)}_{jk\ell}. \tag{1.43}$$

$$\tag{1.44}$$

It can be shown that for $X_n \in C_i$ in the training set, discriminative adjustment of the mean vector follows

$$\widetilde{\mu}^{(i)}_{jk\ell}(n+1) = \widetilde{\mu}^{(i)}_{jk\ell}(n) - \epsilon \frac{\partial \ell_i(X_n;\Lambda)}{\partial \widetilde{\mu}^{(i)}_{jk\ell}}\bigg|_{\Lambda=\Lambda_n} \tag{1.45}$$

where

$$\frac{\partial \ell_i(X;\Lambda)}{\partial \widetilde{\mu}_{jk\ell}^{(i)}} = \frac{\partial \ell_i}{\partial d_i}\frac{\partial d_i}{\partial \widetilde{\mu}_{jk\ell}^{(i)}} \tag{1.46}$$

$$\frac{\partial \ell_i}{\partial d_i} = \gamma \ell_i(d_i)(1-\ell_i(d_i)) \tag{1.47}$$

$$\frac{\partial d_i(X;\Lambda)}{\partial \widetilde{\mu}_{jk\ell}^{(i)}} = -\sum_{t=1}^{T}\delta(\overline{q}_t - j)\frac{\partial \log b_j^{(i)}(\mathbf{x}_t)}{\partial \widetilde{\mu}_{jk\ell}^{(i)}} \tag{1.48}$$

and

$$\frac{\partial}{\partial \widetilde{\mu}_{jk\ell}^{(i)}}\log b_j^{(i)}(\mathbf{x}_t) = c_{jk}^{(i)}(2\pi)^{-d/2}\,|R_{jk}^{(i)}|^{-1/2}\left(b_j^{(i)}(\mathbf{x}_t)\right)^{-1}\left(\frac{x_{t\ell}}{\sigma_{jk\ell}^{(i)}} - \widetilde{\mu}_{jk\ell}^{(i)}\right)$$
$$\cdot\ \exp\left\{-\frac{1}{2}\sum_{\ell=1}^{D}\left(\frac{x_{t\ell}}{\sigma_{jk\ell}^{(i)}} - \widetilde{\mu}_{jk\ell}^{(i)}\right)^2\right\}, \tag{1.49}$$

where γ is the center slope of the exponential sigmoid function for ℓ_i as defined in (1.22), and $\delta(\cdot)$ denotes the Kronecker delta function. Finally,

$$\mu_{jk\ell}^{(i)}(n+1) = \sigma_{jk\ell(n)}^{(i)}\ \widetilde{\mu}_{jk\ell}^{(i)}\ (n+1)\,. \tag{1.50}$$

Similarly, for the variance $\sigma_{jk\ell}^{(i)}$,

$$\widetilde{\sigma}_{jk\ell}^{(i)}(n+1) = \widetilde{\sigma}_{jk\ell}^{(i)}(n) - \epsilon\frac{\partial \ell_i(X_n;\Lambda)}{\partial \widetilde{\sigma}_{jk\ell}^{(i)}}\bigg|_{\Lambda=\Lambda_n} \tag{1.51}$$

where

$$\frac{\partial \ell_i}{\partial \widetilde{\sigma}_{jk\ell}^{(i)}} = -\gamma\ell_i(d_i)[1-\ell_i(d_i)]\cdot\sum_{t=1}^{T}\delta(\overline{q}_t - j)\frac{\partial \log b_j^{(i)}(\mathbf{x}_t)}{\partial \widetilde{\sigma}_{jk\ell}^{(i)}} \tag{1.52}$$

$$\frac{\partial \log b_j^{(i)}(\mathbf{x}_t)}{\partial \widetilde{\sigma}_{jk\ell}^{(i)}} = c_{jk}^{(i)}(2\pi)^{-d/2}\,|R_{jk}^{(i)}|^{-1/2}\exp\left\{-\frac{1}{2}\sum_{\ell=1}^{D}\left(\frac{x_{t\ell}-\mu_{jkl}^{(i)}}{\sigma_{jkl}^{(i)}}\right)^2\right\}$$
$$\cdot\ \left[\left(\frac{x_{t\ell}-\mu_{jkl}^{(i)}}{\sigma_{jkl}^{(i)}}\right)^2 - 1\right]\ \cdot\ \left(b_j^{(i)}(\mathbf{x}_t)\right)^{-1}. \tag{1.53}$$

Finally,

$$\sigma_{jk\ell}^{(i)}(n+1) = \exp\{\widetilde{\sigma}_{jk\ell}^{(i)}(n+1)\}\,. \tag{1.54}$$

Similar derivations for the transition probabilities and the mixture weights can be easily accomplished [18, 51].

As mentioned earlier, the GPD algorithm is a gradient based and unconstrained minimization method. In order to use for discriminant functions from certain families such as probability density functions from HMMs, care must be taken such that those probabilistic constraints are maintained. Transformations in Eqs. (1.40, 1.41, 1.43) are used for this purpose. Another important and perhaps the most difficult issue in GPD based loss function minimization approach is how to design the step size. One apparent reason is that we need a good step size to start with since the model adaptation will be performed only a finite number of times. If the step size is too large, the classifier will be degraded at the start and sequential learning cannot be made successful. If the step size is too small, the convergence speed of the algorithm is too slow and it is practically not useful. The step size problem is related to the particular functional form of the loss function, and to the best of our knowledge, the general solution to it is still lacking. Generally speaking, it should be related to the eigenvalues of the Hessian matrix even though it is an iterative algorithm. For HMM based systems using mixture Gaussian observation densities, parameters in the classifier have different sensitivities to the step size in parameter adaptation. One step size can be too small for some parameters and too large for others. In particular, the magnitude of variances in the mixture Gaussian observation densities can vary in the range between 100 to 10^{-5}. If using a constant step size for all mean vectors, the algorithm will either not converge or will be too slow to become practically useless. The transformation in Eq. (1.42) is critical and provides an effective solution to this problem. In segmental GPD approach, the error rate minimization is performed on the transformed mean vector normalized by its standard deviation. This takes away the dependencies on the variance variations.

During GPD training, the training data can be reused and training can be iterated several times on the same data to reach convergence. Methods from statistical data sampling theory can also be applied here. Instead of sequentially using all training samples, bootstrap resampling schemes or importance of sampling schemes can be used here to extrapolate the sample data distribution or to adapt the classifier towards some specific populations. With the advance of microprocessors, such methods have become computationally feasible. The abovementioned segmental GPD algorithm finds many applications in speech recognition. The recognition performance advantage over the traditional distribution based ML approach is reported from various sites and in different applications [18, 27, 35, 94, 81]. The success of this algorithm in speech recognition provides experimental evidence that classifier design based on error rate minimization is feasible even for dynamic patterns as difficult as speech. Improved recognition performance over traditional distribution based ML approach are also reported in areas outside of speech, such as OCR, image recognition, and handwriting recognition [116].

1.5.5 Relation between MCE and MMI

In addition to the MCE criterion, other criteria are also used in so-called discriminative classifier design. For HMM based systems, the criteria of maximum mutual information (MMI) [4], conditional maximum likelihood estimate (CMLE) [83],

minimum discrimination information (MDI) [33], and H-criteria [38] are other alternatives which have found their use in speech recognition. Among them, MMI is most popular and applied in many applications with success. The MMI approach is based on the mutual information $I(W_c, X)$ between the acoustic observation X and its correct lexical symbol W_c. For the N class classification problem, the logarithm of the mutual information has the following form

$$I(W_c, X) = \log \frac{p(W_c, X)}{p(W_c)p(X)} = \log(\frac{p(X \mid W_c)}{\sum_{k=1}^{N} p(W_k)p(X \mid W_k)}) \\ = r_c(X) - \log(\sum_{k=1}^{N} p(W_k)e^{r_k(X)}) \tag{1.55}$$

where W_k runs over all possible N class symbols, $r_c(X) = \log p(X \mid W_c)$ and $r_k(X) = \log p(X \mid W_k)$ are the log-likelihood scores of X on the correct lexical symbol and the k-th lexical symbol respectively. From Eq. (1.55),

$$\log P(W_c \mid X) = \log(\frac{P(W_c, X)}{P(X)}) = I(W_c, X) + \log p(W_c), \tag{1.56}$$

which relates $I(W_c, X)$ to the posterior probability $p(W_c \mid X)$. In MMI training, the criterion of the classifier design and parameter estimation is to maximize the average mutual information $I(W_C, X)$ on the training set. Experimental speech recognition results indicate that classifier design based on the MMI criterion can lead to better recognition performance than the conventional approach using the ML criterion [84, 114]. Although this criterion is well founded in information theory, possessing good theoretical properties and unique in many ways, it is not based on a direct minimization of the classification error rate and is quite different from the MCE based approach. The relation between MMI and MCE is a very interesting topic and studied in [94, 103]. It is found that under certain conditions, direct comparisons can be made between these two approaches. The discussion below is based on [94]. We derive the explicit relations between MCE and MMI and from there, properties of both approaches can be illustrated.

Let us assume $\{P(W_k) = 1/N \mid k = 1, \ldots, W_N\}$, a case of using a uniformly distributed language model. The mutual information from Eq. (1.55) is given by

$$I(W_c, X) = r_c(X) - \log(\sum_{k=1}^{N} e^{r_k(X)}) + \log N, \tag{1.57}$$

and the MMI model parameter estimation criterion is

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}} E_X(I(W_c, X)) = \underset{\Lambda}{\operatorname{argmax}} E_X((r_c(X) - \log(\sum_{k=1}^{N} e^{r_k(X)}) + \log N). \tag{1.58}$$

The corresponding MCE approach, using the same $r_c(X)$ and $r_n(X)$ as in Eq. (1.55), has the following form:

- The misclassification measure

$$d_c(X) = -r_c(X) + \log(\frac{1}{N-1} \sum_{k:W_k \neq W_c} e^{r_k(X)\eta})^{1/\eta}. \tag{1.59}$$

- The MCE model parameter estimation is

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}}\, E_X(\ell(d_c(X)) \tag{1.60}$$

where

$$\ell(d_c(X)) = \frac{1}{1+\exp(-\gamma d_c(X))} \text{with } \gamma > 0. \tag{1.61}$$

Consider the special case of $\eta = 1$, the following algebraic relations can be derived

$$d_c(X) + \log(N-1) = -r_c(X) + \log(\sum_{k:W_k \neq W_c} e^{r_k(X)}), \tag{1.62}$$

$$e^{d_c(X)+\log(N-1)} + 1 = e^{-r_c(X)}(\sum_{k=1}^{N} e^{r_k(X)}), \quad \text{and} \tag{1.63}$$

$$-\log(e^{d_c(X)+\log(N-1)} + 1) = r_c(x) - \log(\sum_{k=1}^{N} e^{r_k(X)}). \tag{1.64}$$

From Eq. (1.64), the logarithm of the mutual information can be expressed based on the misclassification measure $d_c(X)$ of Eq. (1.61) in the MCE formulation

$$I(W_c, X) = -\log(e^{d_c(X)+\log(N-1)} + 1) + \log N \tag{1.65}$$

$$= \log(\frac{1}{1+e^{d_c(X)+\log(N-1)}}) + \log N \tag{1.66}$$

$$= \log(\ell(-(d_c(X) + \log(N-1))) + \log N. \tag{1.67}$$

From Eq. (1.67), it can be seen that the loss function in MMI is very different from the one used in MCE approach. In addition to a constant shift $\log N$, it is the logarithm of the sigmoid function $\ell()$ on the misclassification measure $d_c(X)$, not on the sigmoid function itself as in MCE approach. The dynamic range of this loss function is from $\log N$ to $-\infty$. It is apparent that the MMI objective function is not trying to approximate the recognition error rate function, and it's optimality to classifier design cannot be directly established from the error rate minimization consideration. From the discriminant function point of view, MMI is to minimize the average misclassification measure $d_c(X)$ if $\log(e^{d_c(X)+\log(N-1)})$ in Eq. (1.65) is approximated by $d_c(X)$. This interpretation is quite different from the posterior probability based interpretation described in Eq. (1.56) which is based on distributional assumptions. It explains the experimental recognition performance improvements obtained from the MMI approach, even in the case where the distributional assumption is known not valid.

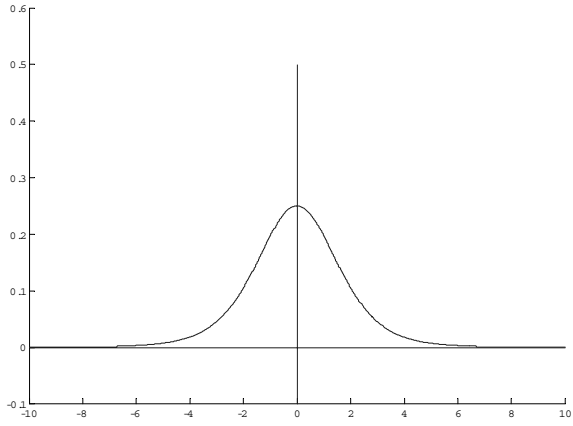

FIGURE 1.1
A plot of the value of the derivative of the sigmoid function.

Based on Eqs. (1.61) and (1.67), the MCE objective function is symmetrical around the misclassification measure $d_c(X)$, whereas the objective function in MMI is asymmetrical. For correct recognition, where $d_c(X) < 0$ and $r_c(X)$ is higher than the average incorrect competitive candidates, both the MCE and MMI objective functions are bounded. They differ though in their sensitivity to the sign changes in $d_c(X)$. The objective function in the MCE approach is directly related to the sign changes in the misclassification measure $d_c(X)$. In the MMI objective function, the sign change in $d_c(X)$ will not lead to a corresponding change in the sign of $d_c(X) + \log(N-1)$ unless $d_c(X)$ is smaller than $-\log(N-1)$. Since N is the number of classes, this can happen in MMI based approach only if $d_c(X) << 0$. When $d_c(X) > 0$, it indicates a recognition error is committed by the recognizer on the random input X. The objective function in the MCE approach is bounded no matter the value of $d_c(X)$. As a contrast, the objective function in MMI is not bounded for $d_c(X) > 0$. This behavior may have some adverse effects in MMI based parameter estimation, since it is based on the mutual information $I(W_c, X)$ averaged over the entire training set. Further insights can be gained by examining the gradient of the objective functions associated with these two approaches. The gradient of the objective function in MCE approach has the following form:

$$\frac{\partial L(W_c, X)}{\partial r_k(X)} = G_k(X)\ell'(d_c(X)) \tag{1.68}$$

with

$$\begin{cases} G_k(X) = \frac{(p(X|W_k))^\eta}{\sum_{k:k\neq c}(p(X|W_k))^\eta} & k \neq c \\ G_c(X) = -1 & k = c \end{cases}$$

where $\ell'()$ is the derivative of the sigmoid function. On the other hand, the gradient of the objective function in the MMI approach based on the misclassification measure $d_c(X)$ is

$$\frac{\partial I(W_c, X)}{\partial r_k(X)} = -G_k(X)\ell(-(d_c(X) + \log(N-1))). \tag{1.70}$$

It can be seen that the gradient function in the MCE approach is based on the differentiated sigmoid function which is concentrated on the class decision boundary $d_c(X) = 0$. The absolute value of the gradient function decreases monotonically if the value of $d_c(X)$ moves away from the decision boundary. On the other hand, the gradient of the objective function in MMI approach is the sigmoid function itself, a function which is monotonic increasing and puts emphasis on extreme false classifications ($d_c(X) \gg 0$). Extreme false classifications are typically outliers. Without proper control, the parameter estimation can be strongly influenced by the outliers, and the estimation results may become biased. This problem can be acute in speech recognition, since such outliers are often from wrong labeling and extreme mismatch in acoustic conditions. Parameter estimation in discriminant function based approach is to find an optimal partition of the sample space such that the recognition error rate can be reduced. Therefore, a robust training algorithm based on the MCE criterion should be more sensitive on the changes in the decision boundaries, since these changes have direct impact on the recognition error rate. It is clear from Eq. (1.68) that this property is embedded in the sigmoid function used in the MCE based approach. Based on Eq. (1.68) and Fig. 1.1, the model parameter adjustments in sigmoid function based MCE approach are modulated by the derivative of the sigmoid function whose value has a peak around $d_c(X) = 0$, the simulated decision boundary.

The MMI and MCE criteria have also been studied experimentally. Speech recognition experimental results of comparing these two approaches are reported by several sites. In particular, a side by side study was given in [94]. In the study, MMI and MCE classifier training were performed based on identical experimental setups and using the same growth-transformation based optimization method for parameter estimation. It was found that both MMI and MCE can lead to speech recognition performance improvements over the ML based approach, and the absolute error rate reduction in the MCE approach is 5.3%, versus 2.5% in the MMI approach. It should be pointed out that although explicit relation between the MMI and MCE approaches can be established using the misclassification measure and under some special conditions, the theoretical study as well as more extensive experimental studies between these two criteria are far from complete and many questions remain to be answered.

1.5.6 Discussions and Comments

The MCE approach described in this section is a discriminant function based approach to pattern classification. The decision rule of the classifier is treated as a discriminant function, and the parameter estimation involves minimizing the expected loss incurred when these decision rules are applied in the classifier. The form of the

loss function is critical in discriminant function based classier design. In the MCE approach, the loss function is constructed in such a way that the recognition error rate of the classifier is embedded in a smooth functional form and minimizing the expected loss of the classifier has a direct relation to the classifier error rate reduction. This direct relation to recognition error rate in the MCE approach has several advantages in classifier design:

- It is meaningful in the sense of minimizing the empirical recognition error rate of the classifier, and this property is not dependent on the parametric form of the discriminant function nor its relation to the form of the true class posterior distribution.
- If the true class posterior distributions are used as discriminant functions, the asymptotic behavior of the classifier will approximate the minimum Bayes risk.

The discriminant function formulation in the MCE approach makes it applicable to various functional decision rules, including those decision rules which are not based on probability functions, such as generalized linear discriminant functions, etc. It also applies to cases where the parametric form of the discriminant function is known to be different from the true class posterior distributions or the discriminant function is selected from other considerations such as mathematical tractability and algorithmic complexity.

The ML based distribution estimation approach to pattern recognition is from a different perspective. The probability distribution (PD) of the random source for recognition is estimated by attributing to the source a parametric model PD and estimating parameters of this PD from given training data. The optimal MAP decision rule is applied to the estimated model PDs as if they were the true probability measures. Such an approach is referred to as the "plug-in" method in statistical literature. Denote the distributions from the source as Q and the distributions from the parametric model as P. The statistics of the source generating the training data are not necessarily those of the models, and the optimality results of the Bayes classifier cannot be applied directly. It is studied in [32] that approaches of ML, MMI can be formulated as cases in the MDI approach, and different assumptions are made about the true PDs of the source to be modeled and the PDs that are used to model the source. The MDI interpretation of the ML approach is that ML estimation of parameters in the model PDs for a given source is equivalent to approximating the empirical distribution of the source on the training data by PDs of the model in the MDI (i.e., Kullback-Leibler distance or relative entropy) sense. In the ML based approach, the MDI measure to be minimized has the following form:

$$MDI_M(X) = \sum_{m=1}^{N} q(m) D(Q_{X|M=m} \parallel P_{X|M=m}) \tag{1.71}$$

where $q(m)$ is the prior probability of the m-th class, and $D(Q_{X|M=m} \parallel P_{X|M=m})$ is the Kullback-Leibler distance between the empirical distribution of the source,

and the distribution of the parametric model estimated from the training data for the given training sample X conditioned on the class label m [32]. Thus a goodness criterion for the ML estimate is introduced. If the PDs of the model include the true PDs of the source, asymptotically it will lead to a Bayes classifier. However, such properties may exist only under a model correctness assumption that the true PDs of the source are covered in the PDs of the parametric model.

If the PDs of the model are rich enough to provide a good approximation to the true PDs of the source, with sufficient training data, the recognition performance may improve as it results in a better approximation to the true PDs of the source through the empirical distribution based on the training data. But if the PDs of the model are quite different comparing to PDs of the source, the achievable recognition performance may be very limited using the distribution estimation approach, and discriminant function based classifier design should be more appropriate. In speech recognition, many assumptions on the model PDs are made regarding the speech generation process. Speech as a signal source for pattern recognition may not be Markovian, nor should it be the case that conditioned on a given state, the observation PDs should be an i.i.d. process. The success of HMMs in speech recognition should not be construed as that the PDs of the model cover the true PDs of the source. In fact, PDs from HMMs are quite limited comparing to the source PDs of speech. Although the nature of source PDs are unknown and no fundamental achievable recognition bounds (similar to Shannon bounds in coding theory) are available, experimental results indicated that discriminant function based MCE approach can lead to significant improvements in recognition performance over the ML based approach. The significance of the MCE approach in speech recognition is twofold. First, a classifier design based on direct minimization of the recognition error rate is a meaningful alternative to distribution estimation based approach. Second, the PDs used in parametric modeling of speech are very limited compared to the true PDs in the source, and the decision rule based on discriminant function approach is a reasonable alternative to the "plug-in" MAP rule which is based on the model correctness assumption.

1.6 Embedded String Model Based MCE Training

In the above mentioned development of the MCE training formalism, the utterance observation X is assumed to be from one of the M classes. For recognition of continuous speech or for speech recognition using subword model units, X is a concatenated string of observations belonging to different classes. For example, a sentence is a sequence of words, each of which is to be modeled by a distribution. The decoding process in continuous speech recognition is to compare (implicitly) all possible (word or subword) string models, and the word string whose string model has the highest likelihood score is chosen as the decoded string. The likelihood score of

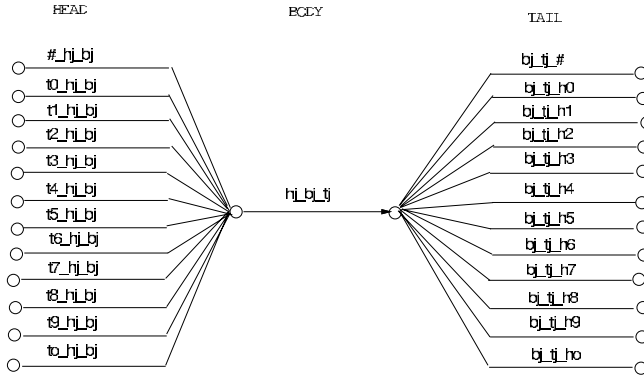

FIGURE 1.2
A structure diagram of a context dependent head-body-tail digit model in speech recognition.

the word string is typically a combination of scores from various models, including the score from the acoustic model, language model, duration model, etc. The main reason of adopting this type of string model is simply that the basic speech recognition model units, which are used to form string models, can be estimated from a finite amount of available training data. A prohibitive number of word strings can be generated even from a very limited vocabulary, and to individually model each word string is not practical to implement for strings with unknown length. On the other hand, long term language models and context dependent acoustic models are used extensively in speech recognition and provide much higher resolution for classifying allophonic acoustic and linguistic events. The use of these detailed and long term knowledge sources in speech recognition has extended the modeling dependencies beyond the level of individual words to phrase groups or at the whole utterance level. Therefore, new formulations are needed to extend the MCE approach to classifier design in continuous speech recognition. In this section, we first describe a general embedded string model based MCE paradigm for continuous speech recognition, and from there, the MCE training for each component in the utterance based string model can be achieved under this unified framework.

1.6.1 String Model Based MCE Approach

Discriminant functions based on string level modeling are necessary in continuous speech recognition, because the classifier decision rules are based on the whole utterance level global matching. The string model which describes the given word

string, that best matches the input speech utterance, has to be determined by Viterbi alignment process between all possible string models and the input speech utterance $X = \{x_1, \ldots, x_T\}$. For ease of representation, we drop the non-acoustic parts in the string model first and consider them separately later.
The string model for a given word string S in an HMM based speech recognition system using continuous observation densities is given by

$$\bar{S}_Q = \underset{S_Q}{\operatorname{argmax}} \log f(X, \Theta_{S_Q}, S_Q \mid \Lambda) \tag{1.72}$$

where S_Q is a possible string model for word string S, Θ_{S_Q} is the optimal state sequence in the string model of S_Q, Λ is the model set of all recognition model units, and $\log f(X, \Theta_{S_Q}, S_Q \mid \Lambda)$ is the log-likelihood score along the optimal state sequence Θ_{S_Q}. In the embedded string model based MCE training described in [19], the discriminant function at the string level is based on the string model for the correct word string and the string models of the N most confusable word strings obtained using a fast tree-trellis N-best search [104]. Let $S = W_1, \cdots, W_{l_s}$ be an arbitrary word string. Given the model set Λ, the optimal state sequence Θ_S is a function of the observation X, and the word string S is often determined by a Viterbi decoding process. The top N best string hypotheses $\{S_1, \cdots, S_N\}$ can be defined inductively as follows,

$$S_1 = \underset{S}{\operatorname{argmax}} \log f(X, \Theta_S, S \mid \Lambda), \tag{1.73}$$

$$S_k = \underset{S \neq S_1, \cdots, S_{k-1}}{\operatorname{argmax}} \log f(X, \Theta_S, S \mid \Lambda). \tag{1.74}$$

The discriminant functions, for $k = 1, \ldots, N$, are

$$g(X, S_k, \Lambda) = \log f(X, \Theta_{S_k}, S_k \mid \Lambda), \tag{1.75}$$

where S_k is the k-th best string, Λ is the HMM set used in the N-best decoding, Θ_k is the optimal path (state sequence) of the k-th string given the model set Λ, and $\log f(X, \Theta_{S_k}, S_k \mid \Lambda)$ is the related log-likelihood score on the optimal path of the k-th string.
For the correct string S_{lex}, the discriminant function is given by

$$g(X, S_{lex}, \Lambda) = \log f(X, \Theta_{S_{lex}}, S_{lex} \mid \Lambda), \tag{1.76}$$

where S_{lex} is the correct string, Θ_{lex} is the optimal alignment path and $\log f(X, \Theta_{lex}, S_{lex} \mid \Lambda)$ is the corresponding log-likelihood score. These discriminant functions are embedded in the MCE based loss function through the following steps:

1. The misclassification measure in embedded string model based MCE training is defined as

$$d(X, \Lambda) = -g(X, S_{lex}, \Lambda) + \log\{\frac{1}{N-1} \sum_{S_k \neq S_{lex}} e^{g(X, S_k, \Lambda)\eta}\}^{\frac{1}{\eta}}. \tag{1.77}$$

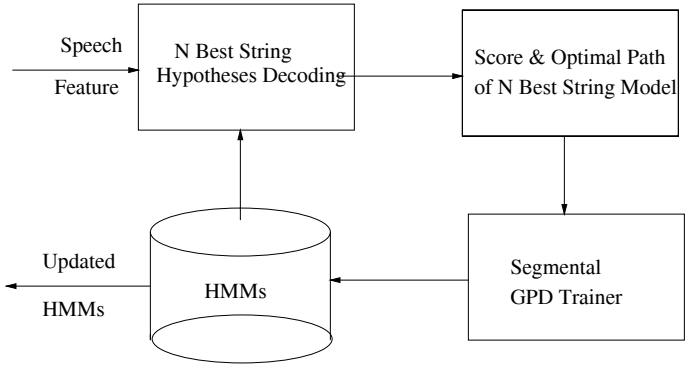

FIGURE 1.3
A diagram of the embedded string model based MCE training process.

2. The loss function in minimum string error rate training is defined as

$$l(X, \Lambda) = \frac{1}{1 + e^{-\gamma d(X,\Lambda)}}, \tag{1.78}$$

where γ is a positive constant, which controls the slope of the sigmoid function.

3. The expected loss which is associated with the string error rate is given by

$$L(\Lambda) = E_X[l(X, \Lambda)]. \tag{1.79}$$

It should be noted that the situation in continuous speech recognition is quite different from the finite M class classification problem where a fixed set of discriminant functions can be pre-specified. The discriminant functions in the embedded string model based MCE approach are dynamic, depending on the particular lexical word string S, random input X, and a list of the N most competitive string models. The N most competitive string models also depend on the string level model matching of the utterance X against the current model set Λ. In the ML based distribution estimation approach, the model parameters are estimated only from the training data with the correct string model. The discriminative information existing in the competing string models is generally not used. The use of the sequential training procedure based on GPD algorithm for parameter adaptation also makes this training procedure "segmental" in the sense that the state segmentation of the speech utterance is used to update the current model, and the updated model is used to introduce new segmentation for the next training sample.

One of the issues in acoustic modeling is how to model the word strings that are not in the training set. In continuous speech recognition, the coverage of the training material on the possible word strings is always limited, given the fact that a huge number of word strings can occur in the language. These unseen strings are in general very hard to model, and ML estimation is based on the seen training data and cannot cover the cases which are unseen. The use of competing string models in MCE training provides a better coverage of word strings, since many of them may not actually occur in the training data. Those confusable word strings are selected based on their confusibility with the correct lexical string given the current model set. They are used to form the string model based discriminant measure and modeled in the smooth MCE based loss function which relates to string error rate. A diagram of the embedded string model based MCE training is given in Fig. 1.3

The embedded string model based MCE approach is well suited for acoustic modeling using detailed context dependent models, where separate acoustic model unit is used to model phoneme with different left and right context. It can describe various long span left and right context dependencies such as triphone, quinphone, etc. One example of a cross-word context dependent model used in connected digit recognition is depicted in Fig. 1.2, which has a full expansion for all possible left and right contexts at the word boundaries. The introduction of the embedded string model in MCE training has two advantages:

- It extends the MCE based discriminant function approach to continuous speech recognition, where modeling each individual word string class is not feasible.
- It provides an exact emulation of the classifier in continuous speech recognition and embeds the utterance level speech manifestation in the basic recognition model units.

In embedded string model based MCE approach, the long term dependencies are embedded in the basic speech recognition model units even if their original context dependency definitions are not. It is observed in the experiments that many monophone based context independent model units obtained from the MCE approach exhibit speech recognition performance of context dependent model units [35]. The embedded string model based MCE approach found applications in various recognition tasks and significant error rate reduction were observed [19, 20, 35, 81, 51, 27]. Although the string model based approach is the natural choice for string error rate minimization, it is possible to include word level error effects in MCE training. One modification proposed in [13] uses word error counts as the weights between the correct lexical string model and the most confusable string model in the misclassification measure. Let the lexical string model be S_{lex} and the most confusable string model S_1, the misclassification measure with the word error count weighting has the following form

$$d(X,\Lambda) = LD(S_{lex}, S_1)(-g(X, S_{lex}, \Lambda) + g(X, S_1, \Lambda)) \tag{1.80}$$

where $LD(S_{lex}, S_1)$ is the so called Levenshtein-distance between the correct reference word string S_{lex} and the rival word string S_1, i.e. the number of errors contained in S_1. The rest of this MCE formulation follows the embedded string error

based MCE approach. However, it should be noted that multiplying a positive constant on the misclassification measure $d(X, \Lambda)$ does not change its sign nor the string error based MCE formulation. When $\gamma \to \infty$, the loss function will still converge to the string error count function, although a word error based weighting is applied.

1.6.2 Combined String Model Based MCE Approach

As mentioned at the beginning of this section, the final decision in speech recognition is based on the combination of scores from various knowledge sources represented by different models. Assuming independence of each model, the final score in the logarithm domain becomes a sum of log-likelihood scores from each individual model. In particular, in addition to the acoustic model, if a language model is used and its score is weighted by a weighting factor λ_L, the final likelihood score of a candidate string is

$$\log f(X \mid W) = \log f(X \mid W) + \lambda_L \log P(W). \tag{1.81}$$

If the model correctness assumption is valid, the log-likelihood score should strictly follow Eq. (1.14) and the score weighting factor $\lambda_L = 1$. However, in speech recognition experiments and applications, it is found that a value of λ_L with $\lambda_L \neq 1$ demonstrates much better recognition performance [63], an indication that the true distribution of the signal source departs from the assumption made by the model. The language model factor λ_L in speech recognition is often tuned and adjusted based on the recognition results on the training and development data. The actual value of the language model factor used in recognition is quite different from the one derived from the model correctness assumption. The tuning procedure itself, being empirical, is a departure from the distribution based pattern recognition approach, and the score combination can be considered as a problem of selecting discriminant functions in pattern classification. The introduction of embedded string model makes it possible to extend the discriminant function based approach to the level of handling multi-model combinations and to the paradigm of combined string model training. The combined string model can be done in the combination of the following two directions. One is horizontal, scores from multiple models and different knowledge sources are combined to form the final score, where each individual model may be estimated separately based on different estimation methods, including using different training data and constraints. Discriminative model combination [13] falls in this category. Another important direction for model combination is to estimate the individual model parameters in the combined string model as an integrated component of the final combined string model. Discriminative feature extraction [58], discriminative language model estimation [21, 113], and embedded string model based estimation using multiple knowledge sources [22] are such approaches, in which the discriminant function is constructed at the combined string model level, and the estimation of parameters at each individual model is achieved by tracing down the model combination tree to each of its leaf nodes following a chain rule like relationship. Although it can be computationally demanding to estimate all model parameters in such a global

manner, the combined string model based approach nevertheless provides an exact characterization of the decision process for even the most sophisticated recognition applications, and it is applied successfully in many speech recognition systems. In order to reduce the computational complexity, model training can be done in a selective way where some portion of the combined string model is assumed fixed while estimating parameters in other selected components of the combined string model. The training process is often iterated several times on the training data, where different model components are selected at each iteration [22]. This integrated approach will be further exemplified in the following subsections.

1.6.2.1 Discriminative Model Combination

The ability to combine multiple models from various knowledge sources in speech recognition is important. This is because speech is a complicated source and can be affected by many factors such as context, prosodics, vocal tract length, ambient environment, speaking style, mode of the speaker, accent, etc. Multiple signal sources from multiple signal bands are also used in speech recognition [79]. Many of these models or knowledge sources may not be based on probabilities, and a discriminant function based approach is a suitable choice for model combination. Let $\{\Lambda_1, \ldots, \Lambda_M\}$ be the individual model components in the model combination. We use the notation $G(X \mid \Lambda_1, \ldots, \Lambda_M)$ to denote the combined string model given random input X, where G is the function selected for model combination. If G is linear

$$G(X \mid \Lambda_1, \ldots, \Lambda_M) = \sum_{k=1}^{M} \lambda_k g(X \mid \Lambda_k), \tag{1.82}$$

where $g(X \mid \Lambda_k)$ is the score from the k-th model, and λ_k is the model combination weights. Discriminative model combination based on the MCE approach is to embed the combined string model based discriminant function $G(X \mid \Lambda_1, \ldots, \Lambda_M)$ in the loss function and estimate the model combination factor $\lambda_1, \ldots, \lambda_M$ as parameters in the combined string model. In particular, the misclassification measure in the combined string model based MCE approach is

$$d(X, \Lambda) = -G(X, S_{lex}, \Lambda) + \log\{\frac{1}{N-1} \sum_{S_k \neq S_{lex}} e^{G(X, S_k, \Lambda)\eta}\}^{\frac{1}{\eta}}, \tag{1.83}$$

and the loss function is defined as

$$l(X, \Lambda) = \frac{1}{1 + e^{-\gamma d(X, \Lambda)}}, \tag{1.84}$$

where γ is a positive constant, which controls the slope of the sigmoid function. To determine model combination coefficients, many optimization methods can be applied to estimate $\lambda_1, \ldots, \lambda_M$ which minimize the expected loss. The popular GPD algorithm has a very simple form in this case [13]. Constraints on the value of model combination coefficients can also be applied during parameter optimization, depending on the nature of the knowledge sources used in the combined string

model. Since estimating the model combination weights is a relatively simple constrained optimization problem, methods of linear programming, conjugate gradient search, etc. become computationally applicable. Discriminative model combination is applied in many applications under the name of combined string model [22, 21], discriminative model combination [13], and universal stochastic engine [41]. The MCE based discriminant function based approach provides a goodness criterion for estimating and adjusting those "tuning parameters" in speech recognition, especially when either a model correctness assumption is not valid or a unified framework is needed to combine knowledge sources that are different in origin or nature. It should be pointed out that discriminative model combination is very different from methods used to combine results from multiple recognizers, such as ROVER [34]. In ROVER, it is based on a voting scheme, and it utilizes the diversity of the recognition errors from independent recognition systems to improve the recognition performance. Discriminative model combination combines different knowledge sources into one discriminant function. The component in the combined string model may not be an independent recognizer, and it can be any knowledge source related to the random input X. However, if each component in the combined string model is a recognizer, both methods apply, and it is an interesting research topic to see how to integrate them together towards a more discriminative combination based on outputs from multiple recognition systems.

1.6.2.2 Discriminative Language Model Estimation

Language modeling is a critical component in speech recognition, and from the source and model point of view, it provides the language level modeling of the source. Moreover, a lot of words in speech are acoustically similar and some of them (homophones) are even identical, such as *"too"* and *"two"*. If only based on acoustic information, identification of these words and phrases in continuous speech can be very difficult, and other knowledge sources, in particular a language model, are needed. A statistical based n-gram language model is a popular choice in speech recognition, and it has the following form $P(w_k \mid w_{k-1}, \ldots, w_{k-n+1})$ which is the estimated probability of observing word w_k given the past $n-1$ word history.

Because the number of possible n-gram probabilities grows exponentially with the order n, lower order language models, such as unigram, bigram, trigram, and four-gram, are used in various speech recognition tasks. The n-gram statistical language model is typically estimated from a large text corpus independent of other knowledge sources in speech recognition. Even with a very large text collection, only portions of unique trigram and four-gram entries can be estimated due to sparseness of the training data. Many language model entries rarely occur in the corpus and back-off schemes (e.g., Katz back-offs [53]) are used to substitute the unseen language model entries with their lower order back-off counterparts. The sparse data problem is a serious issue in language model estimation, and in order to generate more entries, the sample count cut-off threshold in language model estimation is usually set very low, making the estimate far from being reliable. It is also obvious from the language point of view that although the statistical n-gram language model is quite successful

in speech recognition, it is not the "true" source model for speech. It is obvious that human language has a much more complicated structure that often cannot be covered by the simple structure of a fixed n-gram language model.

One direction of the combined string model based MCE approach is discriminative language model estimation. Instead of treating language model estimation as a distribution estimation problem, it is proposed in [21] to formulate the language model estimation as a problem of MCE training according to the combined string model. By expanding the language model part in Eq. (1.82), parameters in the language model can be embedded in the combined string model based MCE formulation, and their values can be estimated based on the minimization of the expected loss. This approach differs from the conventional distribution based ML estimation for language models. First, it is a discriminant function based approach and based on minimization of a special loss function which relates to recognition error rate. Second, the language model parameters become an integral part of the combined string model, not isolate components derived from word occurrence frequencies. It is conditioned on other knowledge sources, such as the acoustic model, duration model, etc.

Many constrained optimization methods can be applied to language model estimation in the combined string model based approach to maintain probability constraints, including transform based GPD adaptation, growth transform based optimization, etc. In the combined string model based approach, language model estimation depends on utterance level string model matching of all model components. The N most confusing string models are determined under the global combined string model framework. Fast word graph search can be used to speed up the search process of preparing competing string models [106]. Sparse training data is an issue which deserves special attention. In order to alleviate the sparse data problem, discriminative language model training can be focused on terms with significant occurrences or language model parameters which occur often such as language model back-off weights, etc. [98, 113]. Discriminative language model estimation can also be used in applications where language models are used as classifiers, such as call routing [96] and dialogue state identification [113]. Experimental results indicate that discriminative language model estimation can improve language model based classifiers comparing to ML based language model estimation [98, 113].

1.6.3 Discriminative Feature Extraction

Most speech recognition systems use some type of spectral analysis on the windowed raw speech waveform. Speech is represented as a sequence of short-time power spectra or related recognition feature vectors. The two types of spectral analysis methods most frequently employed are filter bank analysis and linear predication. Filter bank approaches typically use a bank of bandpass filters. The frequency spacing of the filters are either uniform spaced or critical-band-spaced following Bark scale or Mel scale. These filters are generally highly overlapped and cover the relevant frequency range of the input signal. Time and frequency resolution is an important factor in the filter bank design. Spectrum intensity is often scaled logarithmically and the idea of spectrum weighting is also used to control the feature sensitivity. However, most

feature extractions are based on the analysis of human hearing capabilities, and thus are not necessarily applicable to statistically based machine recognition.
The goal of discriminative feature extraction is to accomplish the speech recognition feature extraction from the standpoint of minimizing the recognition error rate for classification by machines. In place of the Bark scale from hearing, a new frequency scaling [93] can be derived based on the classifier implemented by machines, and other operations in feature extraction can be made discriminative based on the combined string model MCE paradigm. Since these speech recognition feature vectors are part of the combined string model, a goodness criterion of this approach can be derived from the relation of minimization of the expected loss and recognition error rate in the MCE formulation.
One application of discriminative feature extraction is in the design of cepstral lifters [14]. Consider a sequence of speech recognition feature vectors based on the cepstral vectors extracted from a short-time spectral analysis of the speech waveform. It is well known that phoneme class identity useful to speech recognition exists locally in the low frequency cepstral coefficients. Therefore, speech recognizers selectively use this narrow region of cepstral components by applying cepstral weighting or littering based on a windowing function or a lifter $w(n)$. The design of cepstral lifter is to control the non-information bearing cepstral variabilities in order to perform reliable discrimination of sounds. One popular type of the lifter for cepstrum features is based on a raised sine function of the form

$$w(n) = \begin{cases} 1 + h \sin(\frac{n\pi}{L}) & \text{for } n = 1, \ldots, L \\ 0 & \text{for } n \leq 0, k > L, \end{cases}$$

where h is usually chosen as $L/2$, and L is typically $10 - 16$ for speech of 4kHz bandwidth. The weighted feature sequence $w(n)c_n$ from the cepstral lifter, corresponds to a smoothed log power spectrum. The justification of this type of lifter for cepstral feature vectors is given in [46]. In discriminative feature extraction, lifter design can be done according to the combined string model formulation, and instead of relying on human hearing capabilities for choosing the right lifter, the lifter parameters can be estimated using the MCE criterion to minimize the recognition error rate. Speech experiments using discriminative feature extraction based lifter were performed in several tasks [14, 58].

1.7 Verification and Identification

Speaker verification and identification based on voice is an important area in speech research and has been studied for several decades. The general problem of pattern verification can be formulated as follows: given a random input signal X, we want to verify if the signal X is from a signal source S_0. In making a decision regarding the origin of the signal source, two types of errors can occur. One could mistakenly

decide that X is not from the signal source S_0, while the true source of the signal is S_0. This type of error in verification is referred to as a type I error, the error of false rejection or missed detection. The second type of error is that X is accepted as coming from the signal source S_0 while the true source of signal is not S_0. This type of error is often referred to as a type II error or the error of false acceptance. The performance of a verification system is typically evaluated based on the combination of type I and type II errors. The problem of verification can be conveniently formulated into a statistical hypothesis testing problem: given the test signal X, we want to test the null hypothesis, H_0, against the alternative hypothesis, H_1, where H_0 assumes that X is from the source S_0, and H_1 assumes that X is generated by another source S_1. In many applications, the alternative hypothesis H_1 assumes only that X is not generated from the known source S_0, and in such situations, H_1 is a composite hypothesis as opposed to being a simple hypothesis.

In general, a test procedure divides the signal space S_X into two regions R_X and $A_X = S_X - R_X$, and we reject H_0 if $X \in R_X$ and accept H_0 if $X \in A_X$. R_X is often referred to as the critical region of the test. The probabilities of these two types of errors can be expressed as

$$\alpha = P(E_1) = P(X \in R_X \mid H_0), \tag{1.86}$$

and

$$\beta = P(E_2) = P(X \in A_X \mid H_1) = 1 - P(X \in R_X \mid H_1). \tag{1.87}$$

The power of the test, which is an important quantity to characterize the test, is given by

$$\gamma = P(X \in R_X \mid H_1). \tag{1.88}$$

In statistical hypothesis testing, one is often interested in finding the critical region R_X such that the power of the test is maximized, or in other words, the type II error is minimized, at a given level of type I error. A test which is optimal in this sense, is often referred to as the most powerful test. There are plenty of studies available in the statistical literature regarding the design of the optimal tests if $P(X \mid H_0)$ and $P(X \mid H_1)$ are known and fall into some specific distributions, such as the exponential family [65]. In practice, the test procedure is often based on a test statistics $T(X)$ such that H_0 is rejected if $T(X) \geq k$. According to the Neyman-Pearson lemma, $T(X)$ can be based on probability ratio test $T(X) = P(X \mid H_1)/P(X \mid H_0)$ or likelihood ratio test $T(X) = f(X \mid H_1)/f(X \mid H_0)$, and k is selected such that the level of the type I error $P(T(X) \geq k \mid H_0) = \alpha$. However, in most practical verification problems, we have no exact knowledge regarding the distributions of null and alternative hypotheses. This problem is even more acute for the speech signal, which is nonstationary, and the exact nature of speech generation process is still largely unknown. Moreover, parameters of the speech model are estimated from very sparse data points collected from known sources. With the correctness of the model in question and the estimation errors due to sparse training samples for the parameters of the model, the optimality of the test in the classical sense cannot be realized, and discriminant function based methods can be used to improve the

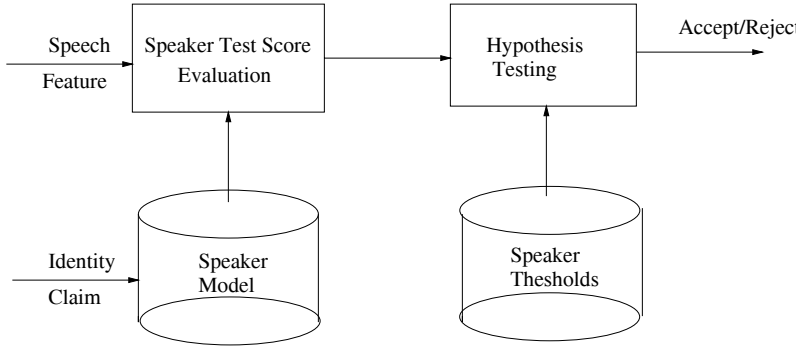

FIGURE 1.4
Block diagram of a speaker verification system

verification performance [74, 110, 107]. In the following subsections, more detailed discussions are given to these approaches.

1.7.1 Speaker Verification and Identification

Authentication by voice has various applications in human-machine communication. Depending on application requirements, it can be classified into two categories, namely speaker verification and speaker identification. Speaker verification involves verifying the identity of a claimed speaker from a known speaker population, and speaker identification involves identifying an unknown speaker from a known population. A typical speaker verification system is shown in Fig. 1.4. Given a sequence of speech feature vectors X and the claimed speaker identity I, the test statistics score $T(X \mid \Lambda_I)$ can be computed from the corresponding speaker model Λ_I. The test score is then compared with a threshold τ_I associated with the claimed speaker to decide if the claimed identity should be accepted or rejected.
Speaker modeling is the most critical part of a speaker verification system. Although the speaker's speech production process can be modeled directly based on the physiological structure of the speaker's articulatory apparatus such as the shape of the vocal tract, etc., it is quite difficult in practice to uniquely extract such structural parameters from speech samples. Instead, indirect speaker modeling is often used in which a set of speech models is created for each speaker based on a collection of speaker specific speech training data. These speech models characterize the acoustic manifestation of speech for a given speaker, which can be done based on various criteria and depend on the type of the verification strategies. In order to model the temporal structure in speech, HMMs are the most popular choice for speaker mod-

eling, although other modeling techniques such as VQ codebook, neural networks, etc., can also be used. HMMs are used in speaker verification to model the general trend of the speech from a known speaker, and depending on applications and the availability of annotated training data, it can be based on fixed phrases, broad phonetic classes, whole words, or even subwords.

In speaker verification, the decision is based on the score of the test statistics $T(X)$. For speaker identification problem, the speaker I is identified as the true speaker if

$$I = \underset{q}{\operatorname{argmax}}\, T(X, q). \tag{1.89}$$

For speaker verification, we usually accept the claimed speaker identity I if the test statistics for I-th speaker

$$T(X, I) > \tau_I. \tag{1.90}$$

The test statistics $T(X, q)$ can be directly based on the likelihood score of the speech X in q-th speaker's model $f(X \mid \Lambda_q)$. However, the normalized score function, which is a form of the generalized likelihood ratio test $f(X \mid \Lambda_q)/f(X \mid \Lambda_{\bar{q}})$, gives a much better speaker verification performance [99]. The likelihood $f(X \mid \Lambda_{\bar{q}})$ models the acoustic space which does not belong to speaker q, and is often obtained from the speaker cohort modeling. For each speaker q, a set of speakers, who are most close to speaker q, is called a cohort set C_q, and can be identified from the training data. The likelihood $f(X \mid \Lambda_{\bar{q}})$ is modeled as a function of the likelihood from competing $f(X \mid \Lambda_r)$ in the cohort set C_q. Good speaker verification performance is observed when $f(X, \Lambda_{\bar{q}})$ is approximated by

$$f(X \mid \Lambda_{\bar{q}}) = \underset{r \in C_q}{\operatorname{argmax}}\, f(X \mid \Lambda_r). \tag{1.91}$$

The discriminative function based MCE approach can be applied using a suitable misclassification measure according to the test statistics [74]. For speaker verification, the verification error can also be characterized by a mis-verification measure based on the test statistics $T(X, I)$, in a way similar to the misclassification measure in recognition, and the MCE based discriminant function approach can be directly adapted to minimize the verification error [74, 62]. In fact, the cohort modeling formulation has a close relation to the misclassification measure in MCE approach. Experimental results indicated that the discriminant function based approach to speaker identification and verification can lead to a significant reduction of the overall verification error. The speaker test statistics are much better separated than the ML based distribution estimation approach [74]. The system sensitivity to threshold selection is much reduced and robustness of the system is improved [74, 78].

In addition to addressing the identification and verification problem as a special classification problem, methods of introducing the structure of statistical test in discriminant function based approach are also attempted. Minimum verification error (MVE) training is such an approach [77, 109]. The main difference to MCE approach is the use of two separate loss functions to model two types of errors in hypothesis testing. The details of MVE approach are described below, and it exemplifies

the discriminant function approach to speaker verification and identification. The mis-verification measure, as opposed to the misclassification measure, for the q-th speaker is defined as:

$$d_q(X,\Lambda) = \begin{cases} -g_q(X,\Lambda) + G_q(X,\Lambda) & \text{if } X \text{ from the claimed } q\text{-th speaker} \\ -G_q(X,\Lambda) + g_q(X,\Lambda) & \text{if } X \text{ is not from the claimed } q\text{-th speaker,} \end{cases}$$

where $g_q(X,\Lambda)$ is the log-likelihood score from the claimed q-th speaker model and $G_q(X,\Lambda)$ is the score from the cohort speaker group $C_{\bar{q}}$. The mis-verification measure is embedded in a smooth sigmoid based loss function $\ell(d(X,\Lambda))$ as in the MCE formulation. Two separate loss functions are used to describe the type I and type II errors. The average loss for each type of errors approximates the empirical type I and type II error rate on the training samples.

$$L_1(X,\Lambda) = \frac{1}{N_1}\sum_{i=1}^{N_1} \ell(d(X_i,\Lambda))1(X_i \in \text{Claimed Speaker}) \tag{1.93}$$

and

$$L_2(X,\Lambda) = \frac{1}{N_2}\sum_{i=1}^{N_2} \ell(d(X_i,\Lambda))1(X_i \notin \text{Claimed Speaker}), \tag{1.94}$$

where $1()$ is the indicator function. The overall expected loss of the MVE is given by

$$L(X,\Lambda) = \lambda_1 L_1(X,\Lambda) + \lambda_2 L_2(X,\Lambda), \tag{1.95}$$

where λ_1 and λ_2 are design parameters which control the influence of type I and type II errors in the overall loss function. The model parameter estimation in MVE training is to minimize the expected loss of Eq. (1.95), which relates to the minimization of empirical error rate of type I and type II errors. The goodness of this criterion is justified from the discriminant function approach, and it is meaningful even when the model correctness assumption cannot be established. Various speaker verification experiments are conducted, and the discriminant function based approach demonstrates significant performance advantages over the distribution estimation based approach [74, 36, 62].

1.7.2 Utterance Verification

Utterance verification is to verify the content of the speech utterance against a claimed (or hypothesized) text string. It can be used to verify the speech recognition result and decide whether the decoded word string is reliable and should be accepted with confidence. One new approach in utterance verification is to verify the information content of the speech utterance against some known data stored in the user personal profile, such as birth date, or against a database to which the information content of the utterance provided by the user with claimed identity should match. This approach of verifying the content of the utterance against a known database is called verbal information verification (VIV) [70]. Verbal information verification can be achieved

without the need of collecting speaker specific training data. Utterance verification as a statistical hypothesis testing problem has a close relation to the problem of speaker verification. The discriminant function approach based on MCE or MVE can be applied to utterance verification with the same fashion as it is applied to speaker verification. However, the purpose of utterance verification is to verify the information content of the utterance not the identity of the speaker. As a consequence, instead of using the speaker model, the confidence scores regarding the word content in the utterance are used in utterance verification. There are many ways to form word and string level confidence measure for utterance verification. In the approach described in [110], the word level confidence score is based on the word level likelihood ratio defined as follows:

$$T(X, W_q) = \frac{L(X \mid H_0(W_q))}{L(X \mid H_1(W_q))} \tag{1.96}$$

where $H_0(W_q)$ and $H_1(W_q)$ are null and alternative hypotheses respectively for verifying random input X has word content W_q. The likelihood of $L(X \mid H_0(W_q)) = L(X \mid \Lambda_{W_q})$ is from the model for W_q. The likelihood for the alternative hypothesis $L(X \mid H_1(W_q))$ is from a different model Ψ_{W_q} which is modeled by two separate HMMs $(\Phi_{W_q}, \Omega_{W_q})$. Φ_{W_q} is an HMM trained on all the training data in the cohort set of W_q. It is used to model the composite acoustic space consists of other words except W_q, and it is sometimes called anti-model to word W_q. Ω_{W_q} is another HMM which is a filler model to model non-keyword event. The likelihood of $L(X \mid H_1(W_q))$ based on these two types of HMMs is given as

$$L(X \mid H_1(W_q)) = L(X \mid \Psi_{W_q}) = [\frac{1}{2}(L(X \mid \Phi_{W_q})^\kappa + L(X \mid \Omega_{W_q})^\kappa]^{1/\kappa}, \tag{1.97}$$

where κ is a positive constant. In discriminant function based utterance verification, the mis-verification measure is

$$d(X, W_q) = -\log L(X \mid \Lambda_{W_q}) + \log L(X \mid \Psi_{W_q}). \tag{1.98}$$

The mis-verification measure is embedded in a sigmoid type function of the form

$$\ell(d(X, W_q) = \frac{1}{1 + \exp[-b\gamma d(X, W_q)]} \tag{1.99}$$

where γ is a positive constant controlling the slope of the sigmoid function, and b takes on the value of $+1$ and -1 as follows:

$$b = \begin{cases} +1 & \text{if } W_q \in CR \\ -1 & \text{if } W_q \in MR \\ -1 & \text{if } W_q \in NK \end{cases}$$

where CR refers to the cases that W_q is correctly recognized, MR refers to the cases that W_q is mis-recognized, and NK refers to the cases that the input speech contains no keyword. Based on Eq. (1.99), the discriminant function based MCE and MVE approach can be applied as in the speaker verification. The above approach

can be extended to string level verification and based on subword units enabling the verification process vocabulary independent [108]. String level verification makes the final rejection/acceptance decision on the keyword hypothesis made by the recognizer. Assuming independence, the string level likelihood ratio can be written as a product of subword-level likelihood ratio:

$$T(X, w_1 w_2, \ldots, w_N) = \prod_{k=1}^{N} \frac{L(X_k \mid \Lambda_{w_k})}{L(X_k \mid \Psi_{w_k})}. \tag{1.101}$$

By collecting terms of the numerator and denominator, the mis-verification measure can be given as follows:

$$d(X, w_1, \ldots, w_N) = \sum_{k=1}^{N} \{L(X_k \mid \Lambda_{w_k}) - L(X_k \mid \Psi_{w_k})\} \tag{1.102}$$

where $w_1, \ldots, w_N$ can be based on the words or subwords according to the modeling strategy used in application. The discriminant function based approach can then be applied to verifying word strings. The construction of string level confidence measure is an active research area, and various statistical criteria can apply. One criterion often used in string level confidence measure is the mini-max principle, which is to minimize the maximum risk of accepting each individual class (or word) in the string. Word class dependent weighting can also be used which gives more emphasis to verify salient words and information baring classes. Discussions of forming various types of string level confidence measures are given in [76, 55] and the references cited there.

The introduction of verification process in speech recognition opens a paradigm of applying the discriminant function based approach in speech processing. Instead of using model likelihood scores as is typical, one approach proposed in [60] is to use generalized confidence score for decoding. The generalized confidence scores is formed by integrating various knowledge sources. Examples of such knowledge sources are frame acoustic likelihood, frame acoustic likelihood ratio, phone and word duration penalties, word language probabilities, word insertion penalties, frame energy penalties, prosodic confidence score, etc. A confidence score preprocessor is used to convert the confidence score from each component into a suitable form for combination. In particular, the non-likelihood ratio based knowledge source is converted into the logarithm domain and integrated using a linear combination. The likelihood ratio based knowledge source is first embedded in a sigmoid function to control its dynamic range and then it is linear combined in the logarithm domain with other components. Both discriminative model combination and discriminative utterance verification are applied to integrate confidence scores from different knowledge sources and to estimate model parameters and combination weights in the generalized confidence score, because the generalized confidence score may not be a probability based likelihood or a probability distribution. Moreover, the confidence score components in the generalized confidence score can be based on different level of information, such as frame level, state level, phone level, word level, etc. During

the decoding process, those confidence score components are applied at different times and at different layers of the decoding network according to their specifications. Therefore, the verification of the decoded partial word string can be done at different phone, word, and phrase group junctions according to the verification based likelihood ratio score components in the generalized confidence score. The decoding process of this approach can be performed in a two pass fashion based on a word graph or the N best list obtained from the first pass search, or in a one pass search which applies the generalized confidence score in search directly [60, 66].

Another approach of utilizing verification in speech recognition and understanding is based on a detection and verification strategy [55]. In this approach, a key phrase based detection using a general acoustic phonetic model is performed first. The detected key phrases are processed in a verification module to verify the detected key phrases and eliminate false alarms. Each key phrase is tagged with a semantic tag and the verified key phrases are connected into sentence hypothesis using task-specific semantic knowledge. A stack decoder is then used to search for the optimal hypothesis that satisfy the semantic constraints. The optimal hypothesis from the stack decoder is further verified at the sentence level based on both acoustic and semantic information for the final output. The discriminant function based approach is applied in both key phrase detection and verification to improve the system performance. Details of this approach are given in [55, 54] and the references cited there.

1.8 Summary

In this chapter, we examined the classical Bayes decision theory approach to the problem of pattern classification, and discussed the implied assumptions and issues as it is applied to the speech recognition problem. The classical Bayes decision theory approach transforms the recognizer design problem to a problem of probability distribution estimation. The limitation of the approach, however, comes from the fact that the true form of the probability distributions of speech signal is realistically unknown, and any assumed distribution form used in the model will deviate from the true one of the source, resulting in suboptimal recognition performance and making the minimum error probability as suggested by the Bayes approach unattainable.

In light of this limitation, the discriminant function based MCE approach was introduced as an alternative to the distribution estimation based approach in pattern recognition. It takes a discriminant function based statistical pattern classification approach to classifier design. For a given set of discriminant functions, the classifier design is to find a set of parameters which minimize the empirical recognition error rate. This is achieved through a special loss function where minimizing the expected loss relates to the reduction of the recognition error rate. The discriminant function based MCE approach applies to cases where the traditional distribution estimation

based approach does not apply, especially when the family of the discriminant functions encountered in the classifier are not based on probability distributions. The goodness of this approach is justified without the model correctness assumption, and it applies to cases where the model correctness assumption is known to be invalid. We formulated the basic theoretical framework of this approach and discussed its relation to other criteria used in the classifier design. The development of MCE approach has led to a new paradigm in pattern recognition, and it leads to recognition performance advantages over the conventional approach in many applications. We studied various extensions of the MCE approach and provided theoretical justifications as well as implementation details when it was applied to different classification problems in speech and language processing. This chapter is based on new developments in discriminant function based MCE approach during the past ten years. Although attempts were made to provide a snapshot of research in this area, the material covered in this chapter is by no means exhaustive. Research on discriminative methods in pattern classification is a fast moving field with new problems and applications from various directions, and we are just at the beginning of realizing the new potential of this approach in pattern recognition.

Acknowledgement

The author would like to acknowledge the contributions of his past and present collaborators. Most materials presented in this chapter are based on joint publications and discussions.

References

[1] S.-I. Amari, "A theory of adaptive pattern classifiers," *IEEE Trans. on Electronic Computers*, Vol. 16, No. 3, pp. 299–307, 1967.

[2] S.-I. Amari, "Learning patterns and pattern sequences by self-organizing nets of threshold elements," *IEEE Transactions on Computers*, Vol. C-12, No. 11, pp. 1197–1206, November 1972.

[3] A. Gunawardana, "Maximum mutual information estimation of acoustic HMM emission densities," *CLSP Research Note No. 40*, Center for Language and Speech Processing, Johns Hopkins University, June 4, 2001.

[4] L.R. Bahl, P.F. Brown, P.V. deSouza and R. L. Mercer, "Maximum mutual information estimation of HMM parameters for speech recognition," *Proceedings of ICASSP-86*, pp. 49–52, 1986.

[5] L.R. Bahl, P.F. Brown, P.V. deSouza and R.L. Mercer, "Estimating hidden Markov model parameters so as to maximize speech recognition accuracy," *IEEE Trans. Speech and Audio Processing*, Vol. 1, No. 1, pp. 77–83, 1993.

[6] L.R. Bahl, F. Jelinek and R.L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on Pattern and Machine Intelligence*, Vol. PAMI-5, pp. 79–190, 1983.

[7] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, Vol. 41, pp. 164–171, 1970.

[8] L.E. Baum, "An inequality and associated maximization techniques in statistical estimation for probabilistic functions of Markov process," *Inequalities*, Vol. 3, pp. 1–8, 1972.

[9] L.E. Baum and J.A. Eagon, "An inequality with applications to statistical predication for functions of Markov process and to model of ecology," *Bull. Amer. Math Soc.*,, Vol. 73, pp. 360–363, 1967.

[10] L.E. Baum and G. Sell, "Growth transformations for functions on manifolds," *Pacific J. Math.*, Vol. 27, No 2, pp. 211–227, 1968.

[11] Bickel and Doksum, *Mathematical Statistics*, Prentice-Hall,, 1977.

[12] A. Benveniste, M. Metivier, and P. Priouet, *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, 1990.

[13] P. Beyerlein, "Discriminative model combination," *Proc. 1997 Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 238–245, 1997.

[14] A. Biem, S. Katagiri, and B.-H. Juang, "Pattern recognition using discriminative feature extraction," *IEEE Trans. Signal Processing*, Vol. 45, pp. 500–504, 1997.

[15] J.R. Blum, "Multidimensional stochastic approximation methods," Ann. Math. Stat. Vol 25, pp. 737–744, 1954.

[16] P.-C. Chang and B.-H. Juang, "Discriminative template training for dynamic programming speech recognition," *Proc. ICASSP92*, Vol. 1, pp. 493 – 496, 1992

[17] P.-C. Chang and B.-H. Juang, "Discriminative training for dynamic programming based speech recognizers," *IEEE Trans. Speech and Audio Processing*, SAP-1(2):135-143, 1993.

[18] W. Chou, C.-H. Lee, and B. H. Juang, "Segmental GPD training of an hidden Markov model based speech recognizer," *IEEE Proc. ICASSP-92*, pp. 473–476, April, 1992.

[19] W. Chou, C.-H. Lee, and B.-H. Juang, "Minimum error rate training based on N-best string models," *IEEE Proc. ICASSP-93*, Vol. II, pp.652–655, 1993.

[20] W. Chou, C-H. Lee, and B-H. Juang, "Minimum error rate training of inter-word context dependent acoustic model units in speech recognition," *Proc. ICSLP-94*, pp. 439–442, Yokohama, 1994.

[21] W. Chou, B-H. Juang, and C.-H. Lee, "Minimum error rate training of combined string models," U.S. Patent #5, 606,644.

[22] W. Chou, C.-H. Lee, and B.-H. Juang, "Speech recognition based on combined string models," *Proc. DARPA ANN Tech. Program CSR Mtg.*, pp. 65–69, 1992.

[23] W. Chou, "Discriminant-function-based minimum recognition error rate pattern-recognition approach to speech recognition," *Proceedings of The IEEE*, Vol. 88, No. 8, pp. 1201-1223, August 2000.

[24] W. Chou, C.-H. Lee, B.-H. Juang and F. K. Soong, "A minimum error rate pattern recognition approach to speech recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 8 No. 1, pp. 5–31, 1994.

[25] W. Chou and B.-H. Juang, *"Adaptive discriminative learning in pattern recognition,"* Technical Report of AT&T Bell Laboratories, 1992.

[26] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

[27] S.M. Chu and Y. Zhao, "Robust speech recognition using discriminative stream weighting and parameter interpolation" *IEEE Proc. ICSLP'98*, pp. 690–694, 1998.

[28] Hai Do-Tu and Michael Installe "Learning algorithms for nonparametric solution to minimum error classification problem," *IEEE Transactions on Computers*, Vol. C-27, No. 7, 648–659, 1978.

[29] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Soc.* Ser. B. Vol. 39, pp. 1-38, 1977.

[30] J.L. Doob, *Stochastic Process*, John Wiley & Sons, 1953.

[31] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.

[32] Y. Ephraim and L. Rabiner, "On the relation between modeling approaches for speech recognition," *IEEE Transactions on Information Theory*, Vol. 36, No. 2, pp. 372–380, March 1990.

[33] Y. Ephraim, A. Dembo and L. Rabiner, "A minimum discrimination information approach for hidden Markov modeling" *IEEE Transactions on Information Theory*, vol. 35, No. 5, pp. 1001–1013, March 1989.

[34] J. Fiscus, "A post-processing system to yield reduced word error rate," *Porc. 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 347–354, 1997.

[35] M.B. Gandhi and J. Jacob, "Natural number recognition using MCE trained inter-word context dependent acoustic models" *IEEE Proc. ICASSP'98*, pp. 457–461, 1998.

[36] C. M. del Alamo et al., "Discriminative training of GMM for speaker identification" *IEEE Proc. ICASSP'96*, pp. 89–93, 1996.

[37] P.S. Gopalakrishnan, et al., "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Trans. on Information Theory*, Vol. 37, no. 1. pp. 107-113, 1991.

[38] P.S. Gopalakrishnan, et al., "Decoder selection based on cross-entropies," *IEEE Proc. ICASSP'88*, pp. 20–23, 1988.

[39] S. Herman and R. Sukkar, "Joint MCE estimation of VQ and HMM parameters for Gaussian mixture selection," *IEEE Proc. ICASSP'98*, pp. 485–488, 1998.

[40] Q. Huo and C. Chan, "The gradient projection method for the training of hidden Markov models," *Speech Communication*, Vol. 13, pp. 307–313, 1993.

[41] X. Huang, M. Belin, F. Alleva, and M. Hwang, "Unified stochastic engine (USE) for speech recognition," *IEEE Proc. ICASSP-93*, pp. 636–639, 1992.

[42] F. Jelinek, "The development of an experimental discrete dictation recognizer," *Proc. IEEE*, Vol. 73, No. 11, pp. 1616–1624, November 1985.

[43] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. of the IEEE*, Vol. 64, No. 4, pp. 532-556, 1976.

[44] F. Jelinek, R. L. Mercer, and S. Roukos, "Principles of lexical Language modeling for speech recognition," *Advances in Speech Signal Processing*, S. Furui and M. M. Sondhi (eds.) pp. 651 – 699, Marcel Dekker, New York, 1991.

[45] F. Jelinek, *Statistical Methods for Speech Recognition*, The MIT Press, Cambridge, 1997.

[46] B.-H. Juang, L.R. Rabiner, and J.G. Wilpon, "On the use of bandpass littering in speech recognition," *IEEE Trans. Acoust. Speech Signal Processing*, ASSP-35, No. 7, 947–954, 1987.

[47] B.-H. Juang and L. R. Rabiner, "Hidden Markov models for speech recognition," *Technometrics*, Vol. 33, No. 3, pp. 251–272, August 1991.

[48] B.-H. Juang, S. S. Levinson and M. M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains," *IEEE Trans. on Information Theory*, Vol. IT-32, No. 2, pp. 307-309, 1986.

[49] B.-H. Juang and L. Rabiner, " The segmental K-means algorithm for estimating parameters of hidden Markov models," *IEEE Trans. Acoust., Speech & Sig. Proc.*, 38(9): pp. 1639-1641, September 1990.

[50] B.-H. Juang and S. Katagiri, " Discriminative learning for minimum error training," *IEEE Trans. Acoust., Speech & Sig. Proc.*, 40(12): 3043-3054, December 1992.

[51] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. on Speech and Audio Processing*, 5(3), May 1997.

[52] D. Kavensky, "Generalization of Baum algorithm to functions on non-linear Manifolds," *Proc. ICASSP'95* Vol 1. pp. 473-476, Detroit, 1995.

[53] M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoustic., Speech, Signal Processing*, Vol. 35, No. 3, pp. 400–401, 1987.

[54] T. Kawahara, C.-H. Lee, and B.-H. Juang, "Key-phrase detection and verification for flexible speech understanding," *IEEE Transactions on Audio and Speech Processing*, 1998.

[55] T. Kawahara, C.-H. Lee, and B.-H. Juang, "Combining key-phrase detection and subword-based verification for flexible speech understanding," *Proc. ICASSP'97*, pp. 1159-1163, 1997.

[56] T. Komori and S. Katagiri, "Application of a generalized probabilistic descent method of dynamic time warping based speech recognition," *IEEE Proc. ICASSP-92*, pp. 497–500, 1992.

[57] S. Katagiri, C.-H. Lee, B.-H. Juang, and T. Komori, "New discriminative training algorithms based on a generalized probabilistic descent method," *Proc. IEEE-SP Workshop on Neural Networks for Signal Processing*, Princeton, 1991.

[58] S. Katagiri, B.-H. Juang, and A. Biem, "Discriminative feature extraction," in *Artificial Neural Networks for Speech and Vision*, R. Mammone, Ed. London, U.K. Chapman and Hall, 1994.

[59] S. Katagiri, B.-H. Juang, and C.-H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probability descent method," *IEEE Proceedings*, 86(11): pp. 2345-2373, 1998.

[60] M.-W. Koo, C.-H. Lee, and B.-H. Juang, "Speech recognition and utterance verification based on a generalized confidence score," *IEEE Transactions on Speech and Audio Processing*, 1999.

[61] M.-W. Koo, C.-H. Lee, and B.-H. Juang, "A new decoder based on a generalized confidence score," *Proc. ICASSP'98*, May 1998.

[62] F. Kormanzskiy and B.-H. Juang, "Discriminative Adaptation for Speaker Verification," *IEEE Proc. ICSLP'96*, pp. 1744–1747, 1996.

[63] K.-F. Lee, *The Development of the SPHINX System*, Kluwer, 1989.

[64] A. Ljolje, Y. Ephraim, and L. Rabiner, "Estimating hidden Markov model parameters by minimizing empirical error rate," *Proc. ICASSP'90*, pp. 709-712, 1990.

[65] E.L. Lehmann, *Testing Statistical Hypotheses*, Wiley, New York, 1959.

[66] E. Lleida and R. Rose, "Efficient decoding and training procedures for utterance verification in continuous speech recognition," *Proc. ICASSP'96*, May 1996.

[67] C.-H. Lee, B.-H. Juang, W. Chou, and J.J. Molina-Perez, "A study on task-independent subword selection and modeling for speech recognition," *Proc. ICSLP96*, pp. 1816-1819, Philadelphia, 1996.

[68] C.-H. Lee, E. Giachin, L.R. Rabiner, R. Pieraccini, and A.E. Rosenberg, "Improved acoustic modeling for speaker independent large vocabulary continuous speech recognition," *Computer Speech and Language*, Vol. 4, No. 2, pp. 103-127, 1992.

[69] C.-H. Lee, F. K. Soong, and Paliwal Eds., "Automatic Speech and Speaker Recognition," Norwell, MA, Kluwer, 1996.

[70] Q. Li, B.-H. Juang, Q. Zhou and C.-H. Lee, "Verbal information verification," *Proc. EuroSpeech'97*, 1997.

[71] C.-H. Lee, "A tutorial on speaker and speech verification," *Proc. NORSIG'98*, pp. 9-16, June., 1998.

[72] Q. Li and B.-H. Juang, "Speaker verification using verbal information verification for automatic enrollment" *Proc. ICASSP'98*, May 1998.

[73] C.-S. Liu, H.-C. Wang, and C.-H. Lee, "Speaker verification using normalized log-likelihood score," *IEEE Trans. Audio & Speech Proc.* 4(1), Jan. 1996.

[74] C.-S. Liu, C.-H. Lee, W. Chou, B.-H. Juang, and A. Rosenberg, "A study on minimum error discriminative training for speaker recognition," *J. Acoust. Soc. Am.* Vol.97, No. 1, pp. 637–648, 1995.

[75] M.G. Rahim and C.-H. Lee, "Simultaneous ANN feature and HMM recognizer design using string-based minimum classification training of HMMs," *Proc. ICSLP'96*, pp. 1824–1827, 1996.

[76] M. G. Rahim, C.-H. Lee, B.-H. Juang, and W. Chou, "Discriminative utterance verification using minimum string verification error (MSVE) training," *Proc. ICASSP'96*, pp. 3485–3588, 1996.

[77] M. G. Rahim and C.-H. Lee, "String based minimum verification error (SB-MVE) training for flexible speech recognition," *Computer, Speech and Language*, 1997.

[78] M. G. Rahim, C.-H. Lee, and B.-H. Juang, "Discriminative utterance verification for connected digit recognition," *IEEE Transaction on Speech and Audio Processing*, 5(3), 1997.

[79] P. McMahon, N. Harte, S. Vaseghi, and P. McCourt, "Discriminative Spectral-Temporal Multi-Resolution Features for Speech Recognition," *IEEE Proc. ICASSP'99*, pp. 1649–1653, 1999.

[80] E. McDemot and S. Katagiri, "Prototype-based MCE/GPD training for various speech units," *Comput. Speech Language*, Vol 8, pp. 351–368, 1994.

[81] E. McDemot and S. Katagiri, "String-level MCE for continuous phoneme recognition," *Proc. EuroSpeech'97*, Vol 1, pp. 123–126, 1997.

[82] J. M. Mendel and K. S. Fu, *Adaptive, Leaning and Pattern Recognition*, Academic Press Inc., 1970.

[83] A. Nadas, D. Nahamoo, and M. A. Picheny, "On a model-robust training method for speech recognition," *IEEE Trans., on Acoustics, Speech and Signal Processing*, Vol 36, No. 9, pp. 1432–1436, 1988.

[84] Y. Normandin, et al., "High performance connected digit recognition using maximum mutual information estimation," *IEEE Trans. on Speech and Audio Processing*, Vol. 2, no. 2. pp. 229-311.

[85] K. A. Papineni, "Discriminative training via linear programming," *Proc. ICASSP'99*.

[86] K. K. Paliwal, M. Bacchiani, and Y. Sagisaka, "Minimum classification error training algorithm for feature extraction and pattern classifier in speech recognition," *Proc. EuroSpeech'95*, pp. 541–554.

[87] D. Pollard, *Convergence of Stochastic Process*, Springer Series in Statistics, 1984.

[88] D. Povey and P. Woodland, "Improved discriminative training techniques for large vocabulary continuous speech recognition," Proc. IEEE ICASSP-01, Salt Lake City, May 2001.

[89] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, Vol. 77, No.2, 257–286, February 1989.

[90] L.R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, NJ, 1993.

[91] C. Rathinavelu and L. Deng, "Use of generalized dynamic feature parameters for speech recognition: maximum likelihood and minimum classification error approaches," *IEEE Proc. ICASSP'95*, pp. 373–376, 1995.

[92] C. Rathinavelu and L. Deng, "The trend HMM with discriminative training for phonetic classification," *IEEE Proc. ICSLP'96*.

[93] C. Rathinavelu and L. Deng, "HMM based speech recognition using state-dependent, discriminatively derived transforms on mel-warped DFT features," *IEEE Proc. ICASSP'96*, pp. 9–13, 1996.

[94] W. Reichl and G. Ruske, "Discriminative training for continuous speech recognition," *Proc. 1995 EuroSpeech'95* Vol 1., pp. 537-540, Madrid, Sept. 1995.

[95] W. Reichl, "Language model adaptation using minimum discrimination information," *Proc. EuroSpeech'99*, pp. 1791-1794, Budapest, 1999.

[96] W. Reichl and S. Ortmanns, "Integrated natural language call routing," *ESCA Workshop on Interactive Dialogue in Multi-Modal Systems*, Kloster Irsee, 1999.

[97] R.A. Render and H.F. Walker, "Mixture densities, maximum Likelihood and the EM algorithm," *SIAM Review*, Vol. 26, No. 2, pp. 195–239, 1984.

[98] L. Rigazio, J.-C. Junqua, and M. Galler "Multilevel discriminative training for spelled word recognition," *IEEE Proc. ICASSP'98*, pp. 489–493, 1998.

[99] A.E. Rosenberg et al., "The use of cohort normalized scores for speaker verification," *Proc. ICSLP'92*, pp. 599–602, 1992.

[100] H. Robbins and S. Monro, " A Stochastic Approximation Method," Ann. Math. Stat., Vol 22, pp. 400–407, 1951.

[101] R.C. Rose, B.-H. Juang, and C.-H. Lee "A training procedure for verifying string hypotheses in continuous speech recognition," *IEEE Proc. ICASSP'95*, pp. 281–284, 1995.

[102] M. Sugiyama and K. Kurinami, "Minimum classification error optimization for a speaker mapping neural networks," *Neural Network for Signal Processing II*, pp. 233-42, 1992.

[103] R. Schluter and W. Macherey, "Comparison of discriminative training criteria," *IEEE Proc. ICASSP'98*, pp. 493–497, 1998.

[104] F.K. Soong and E.-F. Huang, "A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition," *Proc. ICASSP 91*.

[105] R. Schluter, W. Macherey, S. Kanthak, H. Ney, and L. Welling, "Comparison on optimization methods for discriminative training criteria," *IEEE Proc. EuroSpeech'97*, pp. 15–18, Sept. 1997.

[106] R. Schluter, B. Mueller, F. Wessel, and H. Ney, "Interdependence of language models and discriminative training," *Proc. ASRU'99*, pp. 85–89, Dec 1999.

[107] A.R. Setlur, R.A. Sukkar and J. Jacob, "Correction recognition errors via discriminative utterance verification," *IEEE Proc. ICSLP'96*.

[108] R. Sukkar and C.-H. Lee, "Vocabulary independent discriminative utterance verification for non-keyword rejection in subword based speech recognition," *IEEE Transactions on Speech and Audio Processing*, Vol. 4, pp. 420–429, Nov. 1996.

[109] R. Sukkar, "Subword based minimum verification error (SB-MVE) training for task independent utterance verification," *IEEE Proc. ICASSP'98*, pp. 229–233, 1998.

[110] R. Sukkar, M Rahim, and C.-H. Lee, "Utterance verification of keyword strings using word based minimum verification error (WB-MVE) training," *Proc. ICASSP'96*, pp. 516–519, May 1996.

[111] R. Sukkar, A. R. Setlur, C.-H. Lee, and J. Jacob, "Verifying and correcting string hypotheses using discriminative utterance verification," *Speech Communication*, 22:333-342, 1997.

[112] Ya. Z. Tsypkin "Self-Learning – What is It?" *IEEE Transactions on Automatic Control*, Vol. AC-13, No. 6, pp. 608–612, December 1968.

[113] V. Warnke, S. Harbeck, E. Noth, H. Niemann, and M. Levit, "Discriminative estimation of interpolation parameters for language model classifier," *IEEE Proc. ICASSP'99*, pp. 1223–1227, Mar. 1999.

[114] V. Valtchev, J.J. Odell, P.C. Woodland, and S. J. Young, "Lattice based discriminative training for large vocabulary speech recognition," *IEEE Proc. ICASSP'96*, pp. 605–608, May 1996.

[115] Wu, J. and Q. Huo, "Supervised adaptation of MCE-trained CDHMMs using minimum classification error linear regression," *Proc. ICASSP-2002*, Orlando, Florida, May 2002.

[116] C. Yen, S.-S. Kuo, and C.-H. Lee, "Minimum error rate training for PHMM-based text recognition," *IEEE Transactions on Image Processing*, Vol. 8, No. 8, pp. 1120–1124, 1999.

2

Minimum Bayes-Risk Methods in Automatic Speech Recognition

Vaibhava Goel* **and William Byrne†**

**IBM; †Johns Hopkins University*

CONTENTS

Automatic speech recognition (ASR) systems are beginning to appear in a wide variety of information systems. In automobiles or in miniature cellular phones, ASR allows users to control electronic devices without using intrusive keyboards or keypads. In other applications, such as in searching unstructured audio-visual archives, ASR promises access to information that would otherwise be inaccessible due to the difficulty of searching through thousands of hours of recordings.

When ASR is incorporated into an information system it becomes just one aspect of a complex and interrelated collection of automatic procedures. Overall system performance will be measured not by ASR word error rate, but through task specific evaluation criteria. When used in telephones, for example, a typical goal might be to identify the person the user wishes to call while at the same time ignoring everything else the user might say. In another application, such as audio mining, overall system performance may be judged through precision and recall measures more commonly used in information retrieval than in ASR. Given that different performance measurements are likely to be used for different applications, it is desirable to create ASR systems that are tuned for task-specific criteria. However, the maximum likelihood techniques that underlie the training and decision processes of most current ASR systems are not sensitive to application specific goals. A promising approach towards the construction of speech recognizers that are tuned for specific tasks is known as Minimum Bayes-risk (MBR) automatic speech recognition.

The MBR framework assumes that a quantitative measure of recognition performance is known and that recognition should be a decision process that attempts

to minimize the expected error under this measure. The three components of this decision process are: the given error measure; the space of possible decisions; and a probability distribution that allows the measurement of expected error. While in many practical situations the complexity of these components will prohibit the exact implementation of the optimum MBR decision rule, we will present several computationally tractable algorithmic procedures that can be used to approximate the optimal system. Task-specific MBR recognizers will be compared to more commonly used maximum likelihood recognition systems to show that MBR recognizers can be constructed to yield improved performance under a variety of task specific error measures. We will then discuss generalizations of MBR, with an emphasis on the ROVER system combination procedure. As an application of ROVER and ROVER variants, we will present results in system combination for multilingual ASR.

2.1 Minimum Bayes-Risk Classification Framework

In ASR, an acoustic observation sequence $A = a_1, a_2, ..., a_T$ is to be mapped to a word string $W = w_1, w_2, ..., w_N$, where the words w_i belong to a vocabulary $\mathcal{V}$.

Assume that a language $\mathcal{W}$ is known; for large vocabulary tasks it is usually the set of all word strings over $\mathcal{V}$. This language specifies the word strings that could produce any acoustic data seen by that ASR system. Furthermore, assume that the ASR classifier makes its hypothesis selection from a set $\mathcal{W}_h$ of word strings. This set, called the *hypothesis space* of that classifier, would usually be a subset of the language. In general, the hypothesis space could even be a function of the observation A, say $\mathcal{W}_h^A$. The ASR classifier can then be described as $\delta(A) : \mathcal{A} \rightarrow \mathcal{W}_h^A$.

Let $l(W, W')$ be a real valued loss function that describes the cost incurred when an utterance W belonging to language $\mathcal{W}$ is mistranscribed as $W' \in \mathcal{W}_h^A$. $l(W, W')$ could be the word error rate (WER) measured by weighted Levenshtein distance [26, 34] for a speech transcription task, or some measure of semantic distance between sentences for a speech understanding task.

Suppose the true distribution $P(W, A)$ of speech and language is known; this assumes that the true distribution that describes data encountered in practice is available. It would then be possible to measure classifier performance according to Bayes-risk as

$$E_{P(W,A)}[l(W, \delta(A))]. \tag{2.1}$$

This is the expected loss when $\delta(A)$ is used as the classification rule for data generated under $P(W, A)$. Given a loss function and a distribution, the classification rule that minimizes the Bayes-risk of Equation 2.1 is given by [1]

$$\delta(A) = \underset{W' \in \mathcal{W}_h^A}{\operatorname{argmin}} \sum_{W \in \mathcal{W}} l(W, W') P(W|A). \tag{2.2}$$

While the sum in Equation 2.2 is carried out over the entire language of the recognizer, only those word strings with nonzero conditional probability $P(W|A)$ contribute to the sum. Let $\mathcal{W}_e^A$ denote the subset of $\mathcal{W}$ such that

$$\mathcal{W}_e^A = \{W \in \mathcal{W} | P(W|A) > 0\}. \tag{2.3}$$

Equation 2.2 can now be rewritten as

$$\delta(A) = \underset{W' \in \mathcal{W}_h^A}{\operatorname{argmin}} \sum_{W \in \mathcal{W}_e^A} l(W, W') P(W|A). \tag{2.4}$$

We shall refer to this classifier as the minimum Bayes-risk (MBR) classifier. It makes its hypothesis selection by first computing an *expected loss*

$$S(W') = \sum_{W \in \mathcal{W}_e^A} l(W, W') P(W|A), \tag{2.5}$$

for each word string W' in the hypothesis space. The hypothesis with the least expected loss is then selected as its decision.

Since the observations in $\mathcal{W}_e^A$ serve as the evidence used by the MBR classifier, we refer to $\mathcal{W}_e^A$ as the *evidence space* for the acoustic observations A. Similarly, the distribution $P(W|A)$ that defines the evidence space is referred to as the *evidence distribution*.

We now show that specific loss functions can be defined so that two commonly used classification methods, namely likelihood ratio hypothesis testing and maximum a-posteriori classification, can be derived within the MBR framework.

2.1.1 Likelihood Ratio Based Hypothesis Testing

In hypothesis testing the observation A is classified as belonging to one of two classes: a 'null' class that represents a desired statement about A, and an 'alternative' class that represents negation of the 'null.' For instance, in a speaker verification task, the null class represents the desired speaker and the alternative represents impostors. Similarly, in an utterance verification task, the null class is the desired utterance and the alternative is a set of similar sounding utterances.

Let H_n denote the null class and H_a denote the alternative. The likelihood ratio tests (LRT) for hypothesis testing classifies A according to the following decision rule

$$\delta_{\text{LRT}}(A) = \begin{cases} H_n \text{ if } \frac{P(A|H_n)}{P(A|H_a)} > t, \\ H_a \text{ otherwise.} \end{cases} \tag{2.6}$$

The threshold t is set in an application specific manner; it determines the balance between false rejection and false acceptance.

That the LRT is a special case of MBR classification can be seen by considering an evidence space $\mathcal{W}_e = \{H_n, H_a\}$, hypothesis space $\mathcal{W}_h = \{H_n, H_a\}$, and loss

function

$$l_{\text{LRT}}(X,Y) = \begin{cases} 0 & \text{if } X = H_n, Y = H_n, \\ t_1 & \text{if } X = H_a, Y = H_n, \\ t_2 & \text{if } X = H_n, Y = H_a, \\ 0 & \text{if } X = H_a, Y = H_a. \end{cases} \tag{2.7}$$

Under this loss function, the expected loss (Equation 2.5) of H_n is

$$S(H_n) = t_1 P(A|H_a) P(H_a)$$

and that of H_a is $S(H_a) = t_2 P(A|H_n) P(H_n)$. Therefore, H_n is decided on if $S(H_n) < S(H_a)$, or $P(A|H_n)/P(A|H_a) > t_1 P(H_a)/t_2 P(H_n)$. This is the decision rule of Equation 2.6 with $t = t_1 P(H_a)/t_2 P(H_n)$.

2.1.2 Maximum A-Posteriori Probability Classification

The MAP classifier makes its decision from the evidence space itself by selecting the word string with the highest conditional probability. That is,

$$\delta_{\text{MAP}}(A) = \underset{W \in \mathcal{W}_e}{\operatorname{argmax}} P(W|A). \tag{2.8}$$

The MAP classifier can be derived as an MBR classifier by considering a hypothesis space that is identical to the evidence space and a loss function that assigns equal cost to all, say 1, to all misclassifications. That is, under the 0/1 loss function

$$l_{0/1}(W, W') = \begin{cases} 1 & \text{if } W' \neq W \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

the classifier of Equation 2.4 becomes

$$\delta(A) = \underset{W' \in \mathcal{W}_h}{\operatorname{argmax}} P(W'|A) \tag{2.10}$$

where

$$P(W'|A) = \sum_{W \in \mathcal{W}_e : l_{0/1}(W,W')=0} P(W|A). \tag{2.11}$$

This is the MAP classifier of Equation 2.8.

2.1.3 Previous Studies of Application Sensitive ASR

Risk minimization and application specific minimum cost classification have been well studied and practiced in finance, defense, economics, and various other commercial and non-commercial sectors. However, use of these methods in automatic speech recognition has not been extensive. Early investigations into the minimum Bayes-risk training criteria for speech recognizers were performed by Nadas [31, 32]. Since then, other researchers [30, 23] have used Bayes-risk based criteria in ASR system training. Our focus in this chapter, however, is in minimum-risk classification rather than estimation.

Stolcke et.al. [36] proposed an approximation to a minimum Bayes risk classifier for generation of minimum word error rate hypothesis from recognition N-best lists. Other researchers [5, 27, 4, 38] have proposed posterior probability and confidence based hypothesis selection strategies for word error rate reduction that have been shown to be approximations to the MBR classifiers [13, 11, 10, 27, 15, 4]. These approximations have resulted in significant improvements in system performance and suggest that further work on minimum-risk classifiers for ASR may be beneficial.

While MBR recognizers attempt to provide a task specific hypothesis selection mechanism, parallel efforts have been going on developing task specific recognition techniques by creating better task specific models. Notable among these are keyword spotting [35, 39], phrase detection [24, 25], weighted word error rate minimization [29], and identification of named entities in speech [3, 28].

2.2 Practical MBR Procedures for ASR

The algorithmic implementation of MBR recognizers is difficult for three reasons. First, due to the large vocabulary size in many large vocabulary continuous speech recognition (LVCSR) tasks, the evidence and hypothesis spaces in Equation 2.4 tend to be quite large, even for short acoustic observation sequences. For instance, if there are five words in the utterance and $20,000$ words in the vocabulary, there are $20,000^5$ possible word strings, all of which are allowed under an n-gram language model. Second, the problem of large spaces is worsened by the fact that an ASR recognizer often has to process many consecutive utterances. For example, the data could be gathered over the course of an entire news broadcast or an entire ten minute phone conversation. Consequently, the hypothesis and evidence spaces correspond to all possible word strings over many utterances, making it even harder to perform the search and sum computations of Equation 2.4. Finally, while there are efficient dynamic programming techniques to implement the MAP recognizer, such methods are not yet available for an MBR recognizer under an arbitrary loss function.

In this section we present two implementations of the MBR recognizer: first as an N-best list rescoring procedure [36, 14], and second as a search over a recognition lattice [10, 12]. MBR recognition is made possible in both these procedures by segmenting long acoustic data into sentence or phrase length segments (utterances) and restricting the evidence and hypothesis spaces to manageable sets of word strings. The assumptions involved in such segmentation, as well as the issues relating to the distribution of the loss function over these segments, are discussed in Goel et al. [12].

Before presenting these procedures, a computational issue associated with the use of hidden Markov models (HMM) in the evidence distribution will be addressed.

2.2.1 Summation over Hidden State Sequences

Whereas in the discussion thus far it has been assumed that the true evidence distribution is available, this is not the case in practice. This distribution is obtained by applying the Bayes rule

$$P(W|A) = P(W)P(A|W)/P(A). \tag{2.12}$$

Here $P(W)$ is approximated using a *language model*; it is usually a Markov chain based N-gram model. $P(A|W)$ is usually approximated using a hidden Markov model called the *acoustic model*.

Let $\mathcal{S}$ be the set of all the states in the acoustic HMM $P(A|W)$. Let $\mathcal{X}$ denote the set of all possible state sequences that could generate A. The probability $P(A|W)$ is computed as

$$\begin{aligned} P(A|W) &= \sum_{X \in \mathcal{X}} P(A, X|W) \\ &= \sum_{X \in \mathcal{X}} P(X|W)P(A|X, W). \end{aligned} \tag{2.13}$$

The summation of Equation 2.13 is over all possible hidden state sequences. Even if sequences for which $P(X|W)$ is zero are discarded, this can still be very expensive since the number of distinct hidden state sequences grows exponentially with the number of frames in A.

A computationally feasible alternative is to modify the Equation 2.4 as follows

$$\delta(A) = \underset{(W', X') \in \mathcal{W}_h^A \times \mathcal{X}^A}{\operatorname{argmin}} \sum_{(W, X) \in \mathcal{W}_e^A \times \mathcal{X}^A} l((W, X), (W', X'))\, P(W, X, A) \tag{2.14}$$

where $\mathcal{X}^A$ is a sparse sampling of the most likely state sequences in $\mathcal{X}$.

This rearrangement changes both the evidence and the hypothesis spaces from $\mathcal{W}_e^A$ and $\mathcal{W}_h^A$ to $\mathcal{W}_e^A \times \mathcal{X}^A$ and $\mathcal{W}_h^A \times \mathcal{X}^A$, respectively. It anticipates our search over evidence and hypothesis spaces that contain word strings along with their HMM state alignment information. In addition, it gives us the flexibility of working with loss functions that depend on the state alignment of word strings. Also, in the above we used Bayes rule and ignored the $P(A)$ term which is constant for a given A.

For convenience we use W rather than (W, X), $\mathcal{W}_e$ rather than $\mathcal{W}_e^A \times \mathcal{X}^A$, and $\mathcal{W}_h$ rather than $\mathcal{W}_h^A \times \mathcal{X}^A$ in Equation 2.14, with the understanding that word sequences in hypothesis and evidence spaces contain state alignment information with them. With these changes, Equation 2.14 becomes

$$\delta(A) = \underset{W' \in \mathcal{W}_h}{\operatorname{argmin}} \sum_{W \in \mathcal{W}_e} l(W, W')P(W, A). \tag{2.15}$$

Here A is a single utterance, and $P(W, A)$ are lexical-acoustic joint probabilities derived with state alignment information from an *N-best list* or a *lattice*.

2.2.2 MBR Recognition with N-best Lists

An N-best list is a sorted enumeration of word strings and their associated state alignment, sorted in decreasing order of $P(W, A)$. For example, an N-best list generated in response to an utterance corresponding to "I LIVE IN A RURAL AREA" are presented in Table 2.1.

The most direct approximation of Equation 2.15 is by N-best list rescoring procedures as first proposed for WER minimization by Stolcke et al. [36] and later extended to general loss functions by Goel et al. [14]. In this approach, the evidence and hypothesis spaces are restricted to the N-best lists produced by a recognizer. They are denoted $\mathcal{N}_e$ and $\mathcal{N}_h$, respectively, resulting in

$$\delta(A) \approx \underset{W' \in \mathcal{N}_h}{\operatorname{argmin}} \sum_{W \in \mathcal{N}_e} l(W, W') P(W, A). \tag{2.16}$$

This approximation is particularly easy to implement for arbitrary loss functions. However, the use of N-best lists may in some cases be too restrictive an approximation and search errors may result. Therefore, it is of interest to increase the size of these two spaces to the recognition lattice, i.e., to consider more candidates in the search and the sum.

2.2.3 MBR Recognition with Lattices

In the following we present a multistack prefix tree A^* search algorithm that uses recognition lattices as the hypothesis and evidence spaces. The development of the algorithm proceeds as follows. We start by introducing statistical quantities derived from the lattice that are needed by the search procedure. We then present a single stack A^* search directly over the lattices. This search is further refined by introducing a prefix tree, multistack strategy. For clarity of presentation we formulate the A^* search for minimization of WER. This is realized as a minimum-risk procedure under a loss function based on Levenshtein distance, henceforth referred to as the Levenshtein loss function. We end this section by a discussion of the feasibility of the A^* search for other loss functions.

2.2.3.1 Lattice Definitions

A recognition lattice is a compact representation for a large set of word strings and their time boundaries.* It is an acyclic directed graph $(\mathcal{N}, \mathcal{E}, n_s, n_e, \rho)$; $\mathcal{N}$ is the set of nodes; $\mathcal{E}$ is the set of edges; n_s is the unique lattice start node, n_e is the unique lattice end node; and $\rho : \mathcal{N} \times \mathcal{E} \rightarrow \mathcal{N}$ specifies lattice connectivity. Each node in the lattice is labeled by a word and a time. Each edge has a start node and an end node. Edges are associated with the words at their end nodes and with the

*Lattices are generated using word strings and their state level alignment with the acoustic frames. However, we consider lattices in which the state alignment information is discarded and only the word time boundaries are kept.

time interval from their start node to their end node. They are also labeled by the joint acoustic and language model log-probability that their word occurs during the associated interval. This joint log-probability is conditioned on the history specified by the start node of the edge. For example, in Figure 2.1, an edge identifies the hypothesis that the word NOW begins at 0.6 sec and ends at 1.3 sec. The number -2.3 on this edge is log-probability that the word NOW occurs between 0.6 sec and 1.3 sec., given that the word HELLO is present from the start of the acoustic data until 0.6 sec.

A *path* or *complete path* is a sequence of connected nodes and links from n_s to n_e through the lattice. A *path segment* is a sequence of connected nodes from an internal lattice node n_1 to another internal lattice node n_2; n_1 may be n_s and n_2 may be n_e. A *partial path* is a sequence of connected nodes and links from n_s to an internal lattice node n; it may be a complete path if n is n_e. The acoustic segment corresponding to a path segment W_x from n_x to n_y shall be denoted $A(W_x)$.

Let W_p be a partial path from lattice start n_s to n_x, W_x be a path segment from n_x to n_y, and W_e be a path segment from n_x to the lattice end n_e. The acoustic segments corresponding to these three path segments will be denoted $A(W_p)$, $A(W_x)$, and $A(W_e)$, respectively. The sum of log-probabilities on the edges along W_p gives $\ln P(W_p, A(W_p))$, the sum of log-probabilities along W_x gives $\ln P(W_x, A(W_x)|W_p)$, and the sum of log-probabilities along W_e gives $\ln P(W_e, A(W_e)|W_p)$.

We introduce the *partial path log-probability*, the *lattice backward log-probability*, and the *lattice total probability* of a partial hypothesis W_p as follows. The partial path log-probability of W_p is

$$L_f(W_p) = \ln \{P(W_p, A(W_p))\}. \tag{2.17}$$

The lattice backward log-probability of W_p is

$$L_b(W_p) = \ln \left\{ \sum_{W_e : W_p \cdot W_e \in \mathcal{W}_{lat}} P(W_e, A(W_e)|W_p) \right\}, \tag{2.18}$$

where $\mathcal{W}_{lat}$ denotes the set of all complete paths in the lattice. The lattice total probability of W_p is

$$T(W_p) = \exp \{L_f(W_p) + L_b(W_p)\}. \tag{2.19}$$

Substituting the definitions of $L_f(W_p)$ and $L_b(W_p)$ in Equation 2.19 we get

$$\begin{aligned} T(W_p) &= \exp \left\{ \ln [P(W_p, A(W_p))] + \ln \left[\sum_{W_e : W_p \cdot W_e \in \mathcal{W}_{lat}} P(W_e, A(W_e))|W_p) \right] \right\} \\ &= \exp \left\{ \ln \left[\sum_{W_e : W_p \cdot W_e \in \mathcal{W}_{lat}} P(W_e, A(W_e))|W_p) P(W_p, A(W_p)) \right] \right\} \end{aligned}$$

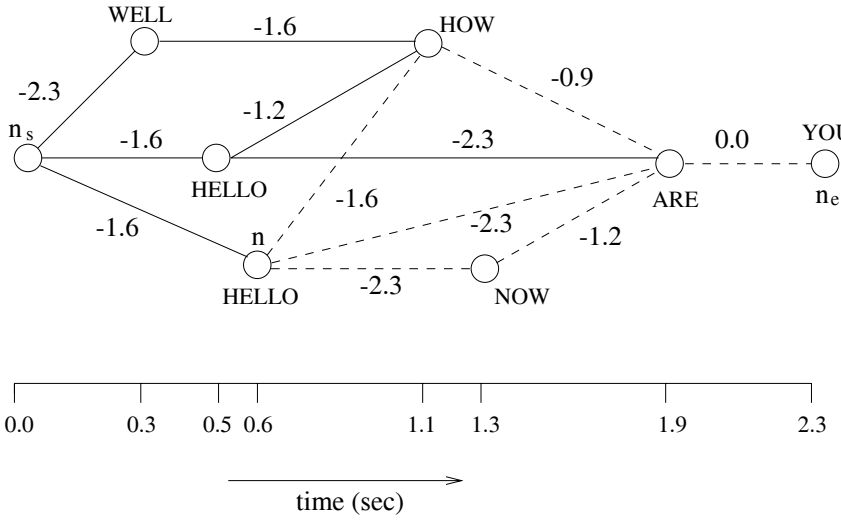

FIGURE 2.1
An example lattice. The time marks correspond to the node times and the word ending times. The numbers on the edges are logarithms of conditional joint probabilities as described in the text. The partial path log-probability of a partial hypothesis is the log of the probability of its path; the partial path W_p = ('HELLO','0.6') in this lattice has value -1.6. The lattice backward log-probability of a partial hypothesis W_p is the log of the sum of probabilities of all lattice paths from end node of W_p to the lattice end node; for the partial path W_p = ('HELLO','0.6') in this lattice these paths are indicated by dotted lines and the lattice backward log-probability of this W_p is -1.55. The lattice total probability of a partial path is the exponentiated sum of its partial path log-probability and lattice backward log-probability; its value is 0.043 for W_p = ('HELLO', '0.6') in the lattice above.

$$= \exp\left\{\ln\left[\sum_{W_e: W_p \cdot W_e \in \mathcal{W}_{lat}} P(W_p \cdot W_e, A)\right]\right\}$$
$$= \sum_{W_e: W_p \cdot W_e \in \mathcal{W}_{lat}} P(W_p \cdot W_e, A)$$

The lattice total probability $T(W_p)$ of a partial hypothesis W_p could therefore be interpreted as the joint probability of observing the acoustics A and all possible complete hypotheses that have the prefix W_p. These probabilities are illustrated in Figure 2.1.

2.2.3.2 A^* Search Under General Loss Functions

The set of all complete paths in the lattice, $\mathcal{W}_{lat}$, constitutes the hypothesis space $\mathcal{W}_h$ of Equation 2.15. It also forms the evidence space $\mathcal{W}_e$ (Equation 2.15); the associated joint log-probability $\ln\{P(W,A)\}$ can be computed by adding the log-probabilities on lattice edges along W. Therefore, on the lattice we would implement

$$\delta(A) = \underset{W' \in \mathcal{W}_{lat}}{\operatorname{argmin}} \sum_{W \in \mathcal{W}_{lat}} l(W,W')P(W,A). \tag{2.20}$$

The goal is to find a complete hypothesis W', i.e., a path from n_s to n_e through the lattice, such that its expected loss

$$S(W') = \sum_{W \in \mathcal{W}_{lat}} l(W,W')P(W,A) \tag{2.21}$$

is the least of all complete hypotheses in the lattice. This search for W' can be effectively implemented as an A^* algorithm [17, 18, 19, 33] which proceeds by extending partial hypotheses forward through the lattice.
Two cost functions are required for the search. The first cost function is associated with each hypothesis W_p, whether partial or complete. Its value is a lower bound on the expected loss (Equation 2.21) that can be obtained by extending the hypothesis through the lattice to completion

$$C(W_p) \leq \min_{W_e : W_p \cdot W_e \in \mathcal{W}_{lat}} \sum_{W \in \mathcal{W}_{lat}} l(W, W_p \cdot W_e)P(W,A). \tag{2.22}$$

The second cost function is only associated with complete hypotheses. It is an over-estimate of the expected loss of a complete hypothesis W'

$$\overline{C}(W') \geq \sum_{W \in \mathcal{W}_{lat}} l(W,W')P(W,A). \tag{2.23}$$

Hypotheses are kept in a priority queue which is sorted by cost C, with the smallest cost hypothesis at the top. We shall use the term "stack" to refer to the queue since in speech recognition the A^* algorithms have historically been presented in terms of stacks [19, 33]. At every iteration the hypothesis at the top of the stack is extended. When there is a complete hypothesis at the top, its second cost $\overline{C}$ is computed. If this over-estimated cost $\overline{C}$ is smaller than the under estimated cost C of the next stack hypothesis, or if there is no partial hypothesis left in the stack, the algorithm terminates. We note that A^* procedures usually employ an exact expected loss (Equation 2.21) for complete hypotheses; however, this is prohibitively expensive to find in our case, therefore we use the over-estimate.

2.2.3.3 Single Stack Search Under Levenshtein Loss Function

We now present usable cost functions for the Levenshtein distance $L(W,W')$. These costs are not unique, and the efficiency of the search depends on the quality of both the under-estimate C and the over-estimate $\overline{C}$.

As a technical aside, we note that the Levenshtein loss function is not sensitive to the word time boundaries present in the lattice. Therefore, the word time boundaries would be summed over during the search. Thus, this A^* search implicitly provides marginalization over different time segments of word strings present in the lattice.
Let $\mathcal{W}_{st}$ denote the set of all complete and partial hypotheses in the stack. The under-estimate for partial hypotheses is

$$C(W_p) = \sum_{\tilde{W} \in \mathcal{W}_{st}} \left\{ \min_{\substack{X : W_p \cdot X \in \mathcal{W} \\ Y : \tilde{W} \cdot Y \in \mathcal{W}}} L(\tilde{W} \cdot Y, W_p \cdot X) \right\} T(\tilde{W}), \tag{2.24}$$

where $\mathcal{W}$ is the set of all possible word strings and their all possible time boundaries that can be constructed by concatenating zero or more words of the vocabulary. The derivation showing that this cost function satisfies Equation 2.22 is presented in Goel et.al. [12].
The over estimate for a complete hypothesis W' can be computed as follows:

- For a hypothesis $\tilde{W}$ in stack $\mathcal{W}_{st}$, let $N(\tilde{W})$ be the length of the longest path from its end node to the lattice end node n_e.
- Append each hypothesis $\tilde{W}$ in the stack by $N(\tilde{W})$ instances of out of vocabulary markers D. These markers do not match any word in the vocabulary.
- Compute the over-estimate

$$\overline{C}(W') = \sum_{\tilde{W} \in \mathcal{W}_{st}} L(\tilde{W} \cdot D_1^{N(\tilde{W})}, W')\, T(\tilde{W}). \tag{2.25}$$

A derivation showing that this estimate satisfies Equation 2.23 is given in Goel et al. [12].
With the under-estimate (Equation 2.24) and the over-estimate (Equation 2.25), the following single stack search algorithm can be used to find the desired hypothesis in the recognition lattice.

1. Mark the lattice nodes by the lattice backward log-probability (Equation 2.18). At each node keep the length of the longest path to the end of the lattice.
2. Maintain a stack $\mathcal{W}_{st}$ of partial and complete hypotheses. Each partial stack entry contains a hypothesis W_p, $L_f(W_p)$ (Equation 2.17), $T(W_p)$ (Equation 2.19), and $C(W_p)$ (Equation 2.24). Each complete stack entry contains a hypothesis W', $T(W')$, $C(W')$, and $\overline{C}(W')$ (Equation 2.25). The stack ordering is defined first by increasing values of $C(\cdot)$, and second by decreasing values of $T(\cdot)$ in cases of identical $C(\cdot)$.

3. Initialize the search by inserting the start node of the lattice, i.e., the NULL hypothesis, into the stack.

4. If there are incomplete hypotheses in the stack, extend the top incomplete hypothesis by all lattice arcs that leave its end node. Compute $C(W_p)$ for each of the newly created partial hypothesis W_p. Compute $C(W')$, and $\overline{C}(W')$ for each newly created complete hypothesis W'. Otherwise, if there are no incomplete stack hypotheses, select the hypothesis with least $\overline{C}(W')$. This is the desired candidate.

5. Update the cost estimates (Equations 2.24 and 2.25) of all other partial and complete stack hypotheses after adding these newly created hypotheses to the evidence space. Insert the newly created hypotheses at their appropriate places (sorted first by $C(\cdot)$ and second by $T(\cdot)$ in case of ties) in the stack. Pruning may be applied during the insertion (see Section 2.2.3.5).

6. If there is a complete hypothesis at the top of the stack and if its over estimate is smaller than the under-estimate of second stack hypothesis (partial or complete), it is the desired candidate and the search ends. Otherwise go to step 4.

2.2.3.4 Prefix Tree Search Under Levenshtein Loss Function

In our treatment so far, the time segmentation of each hypothesis is retained so that hypotheses are distinct if they have identical word content but different time segmentation. Since the Levenshtein distance does not depend on the time segmentation of hypotheses, we can obtain further search efficiency by removing time information from the lattices as follows. Let U be the operator that strips the time segmentations from hypotheses. Given a partial hypothesis W_p from the stack $\mathcal{W}_{st}$, let $\Phi_p = U(W_p)$ be its word contents. Let $T(\Phi_p) = \sum_{\tilde{W} \in \mathcal{W}_{st} : U(\tilde{W}) = \Phi_p} T(\tilde{W})$ be the induced total probability of Φ_p over the current stack. The cost function of Equation 2.24 can be rearranged using the operator U as

$$
\begin{aligned}
C(W_p) &= \sum_{\tilde{W} \in \mathcal{W}_{st}} \min_{\substack{X : W_p \cdot X \in \mathcal{W} \\ Y : \tilde{W} \cdot Y \in \mathcal{W}}} L(\tilde{W} \cdot Y, W_p \cdot X)\, T(\tilde{W}) \\
&= \sum_{\Phi \in U(\mathcal{W}_{st})} \sum_{\tilde{W} : U(\tilde{W}) = \Phi} \min_{\substack{a : \Phi_p \cdot a \in U(\mathcal{W}) \\ b : \Phi \cdot b \in U(\mathcal{W})}} L(\Phi \cdot b, \Phi_p \cdot a)\, T(\tilde{W}) \\
&= \sum_{\Phi \in U(\mathcal{W}_{st})} \min_{\substack{a : \Phi_p \cdot a \in U(\mathcal{W}) \\ b : \Phi \cdot b \in U(\mathcal{W})}} L(\Phi \cdot b, \Phi_p \cdot a) \sum_{\tilde{W} : U(\tilde{W}) = \Phi} T(\tilde{W})
\end{aligned}
$$

$$= \sum_{\Phi \in U(\mathcal{W}_{st})} \min_{\substack{a:\Phi_p \cdot a \in U(\mathcal{W}) \\ b:\Phi \cdot b \in U(\mathcal{W})}} L(\Phi \cdot b, \Phi_p \cdot a)\, T(\Phi)$$

$$= C(\Phi_p) \tag{2.26}$$

Therefore the cost of a partial hypothesis W_p depends only on its word contents Φ_p. This suggests that we can introduce a *prefix tree* as a compact representation of the word sequences associated with all partial hypotheses in the stack. A node in the prefix tree identifies a set of hypotheses and their end nodes in the lattice. Now the search can be performed over the prefix tree. It is the same as the single stack search except that

1. The stack contains prefix tree nodes and is ordered first by $C(\Phi_p)$ (Equation 2.26) and then by $T(\Phi_p)$ in case of ties.
2. The lattice paths corresponding to the prefix tree node at the top of the stack are extended by one word. These extensions yield a new set of prefix tree nodes to be inserted in the stack.

The over-estimate in the prefix tree search is still computed according to Equation 2.25. Due to its dependence on the longest completion of lattice paths, different paths at one prefix tree node would contribute differently to this over-estimate. Other over-estimates that are dependent only on the prefix tree node could be derived; one such estimate could be based on taking the maximum value among the longest completion lengths of all the lattice paths that end at one prefix tree node.
A significant advantage of using prefix trees for the Levenshtein distance is that they facilitate storage and computation of

$$\min_{\substack{a:\Phi_p \cdot a \in U(\mathcal{W}) \\ b:\Phi \cdot b \in U(\mathcal{W})}} L(\Phi \cdot b, \Phi_p \cdot a). \tag{2.27}$$

This quantity, named *partial hypothesis comparison cost*, is needed in Equation 2.26 above. Efficient computation of the partial hypothesis comparison cost is essential for the feasibility and speed of the A^* search. Due to the recursive nature of the Levenshtein distance, the partial hypothesis comparison cost can be computed progressively as the search proceeds.

2.2.3.5 Pruning and Multistack Organization of the Prefix Tree Search

Although the derivations of under estimates and over estimates of costs (Equations 2.24 and 2.25) did not take stack pruning into account, pruning is essential for these algorithms to be feasible [16, 20]. When entries are pruned from the stack, Equation 2.24 is still a valid under estimate but Equation 2.25 is no longer a valid over estimate. It is however a valid over estimate for the sub-lattice of the original lattice that could be constructed by completion of the partial hypotheses in the pruned stack. Therefore, in the search algorithms above, we can at best hope to find the optimal solution within this sub-lattice.

The single stack search (Section 2.2.3.3) and the prefix tree search (Section 2.2.3.4) have the disadvantage that the costs of partial hypotheses of different lengths are compared. This is acceptable under the search formulation, but is not a good comparison for use in pruning since it favors short hypotheses. Thus it may be sub-optimal to prune candidates based on their cost in the single stack. In an attempt to avoid this we use a multistack implementation which is a fairly simple extension of the prefix tree search that maintains a separate stack for each hypothesis length. This multistack organization has been found to have better pruning characteristics in practice. It is this multistack prefix tree A^* search that we report the results on.

2.2.3.6 Loss Functions Other than Levenshtein Distance

From the A^* search formulation of Section 5.6.2 it is clear that the feasibility of the search depends on the ability to compute the two cost functions (Equations 2.22 and 2.23) that provide lower and upper bounds on the expected loss. One such pair of cost functions is provided for the Levenshtein loss function in Equations 2.24 and 2.25. It can be seen from the derivation of the under estimate cost of Equation 2.24 that it is directly generalizable to any arbitrary loss function, if the efficient computation of the prefix comparison cost for that loss function is possible. The computation of an over estimate cost needs to be addressed on a case by case basis.

2.3 Segmental MBR Procedures

We now discuss MBR recognition strategies that reduce utterance level recognition into a sequence of simpler MBR recognition problems. The lattices or N-best lists used as hypothesis and evidence spaces are segmented into sets of words and short phrases which form individual recognition problems that are attacked separately. The solutions of these smaller problems are then joined to produce a single MBR hypothesis for the entire utterance.

This segmental MBR (SMBR) recognition strategy has several advantages relative to utterance level MBR. The segmentation can be performed to identify *high confidence regions* within the evidence space produced by the first-pass ASR system. Within these regions the ASR system was able to produce reliable word hypotheses. SMBR then focuses on the *low confidence regions* in which the first-pass system failed to produce a hypothesis with confidence. The value of this is that search space is expanded where the first-pass system did not perform well and contracted where the initial hypothesis is adequate. We now present a general formulation of these SMBR procedures after which several specific variants will be described.

We first describe the segmentation process. Let R_e be an evidence segmentation rule that uniquely segments each word string in $\mathcal{W}_e$ into N substrings of zero or more words. Applying R_e to $\mathcal{W}_e$ generates *segment sets* $\mathcal{W}_e^i$, $i = 1, ..., N$. These segment sets consist of substrings from the original evidence space. $R_e^i(W)$ denotes the i^{th}

segment of the word sequence W, i.e., $R_e(W) = \left\{R_e^1(W), \cdots, R_e^N(W)\right\}$.
In a similar way, let R_h be a hypothesis segmentation rule coupled with R_e. R_h uniquely divides each string in the hypothesis space $\mathcal{W}_h$ into N segments, and $R_h^i(W')$ denotes the i^{th} segment of the hypothesis W'. The constraint on R_h is that it must have a *conjunction rule* J_h for concatenating strings from segment sets $\mathcal{W}_h^i$. The conjunction rule must be such that when $W' \in \mathcal{W}_h$ is segmented, applying J_h to the segments reproduces W'.
To summarize how the segmentation and conjunction process will be used in decoding, the hypothesis segmentation rule will be used to define N hypothesis sets $\mathcal{W}_h^i$. A single hypothesis W'^i will be chosen from each hypothesis segment set based on the corresponding evidence segment set $\mathcal{W}_e^i$. The conjunction rule will be then used to produce a single utterance level hypothesis $W' = J_h(W'^1 \cdots W'^N)$ from the individual segment hypotheses. It is worth noting that this process of segmention and conjunction may in fact enlarge the original hypothesis space by introducing new hypotheses constructed from substrings taken from the original hypotheses. The enlarged space is adopted in place of the original hypothesis space.
We now describe how the utterance level MBR problem can be reduced to individual MBR recognition problems. This follows from the following assumption concerning the sensitivity of the loss function with respect to the segmentation of the hypothesis and evidence spaces. Assume that the utterance level loss can be found from the losses over the segment sets as

$$l(W, W') = \sum_{i=1}^{N} l^i(R_e^i(W), R_h^i(W')). \tag{2.28}$$

where l^i is a loss function defined on the i^{th} segment set. In effect, we assume that even though segmentation introduces constraints in the alignment between sequences, the overall loss function is not affected.
We can now state the following proposition which follows directly by the substitution of Equation 2.28 into Equation 2.4.
Proposition. An utterance level MBR recognizer of Equation 2.4 can be implemented as a concatenation of N MBR recognizers [13]

$$\delta(A) = J_h\left(\delta^i(A)|_{i=1}^N\right), \tag{2.29}$$

where

$$\delta^i(A) = \underset{W'^i \in \mathcal{W}_h^i}{\operatorname{argmin}} \sum_{W^i \in \mathcal{W}_e^i} l^i(W^i, W'^i) P^i(W^i|A), \tag{2.30}$$

and $P^i(W^i|A)$ is the marginal probability over the i^{th} evidence segment set

$$P^i(W^i) = \sum_{W \in \mathcal{W}_e : R_e^i(W) = W^i} P(W|A). \tag{2.31}$$

Therefore, under the assumption of Equation 2.28, utterance level MBR recognition becomes a sequence of smaller MBR recognition problems.

We note that while the utterance level MBR recognizer is implemented as a sequence of segmental MBR recognizers, the acoustic data is not segmented at all. All evidence originally available is used to compute the marginal probabilities $P^i(W^i|A)$. Also note that there is no assumption of linguistic independence between word strings belonging to adjacent evidence segment sets; the language model spans segments and could even be applied at the entire utterance level.
In practice it may be difficult to segment the evidence and hypothesis spaces so that the loss function distributes according to Equation 2.28. However, given any segmentation, we can identify an associated utterance level *induced* loss function, defined as

$$l_I(W, W') = \sum_{i=1}^{N} l^i(R_e^i(W), R_h^i(W')). \tag{2.32}$$

Clearly, the segmental MBR recognizers are equivalent to an utterance level MBR recognizer under the loss function l_I. The overall performance under the desired loss function l should depend on how well l_I approximates l.

2.3.1 Segmental Voting

A special case of the segmental MBR recognition arises under certain conditions. Suppose each evidence segment set contains at most one word from each evidence word string; each hypothesis segment set contains at most one word from each hypothesis word string; and there is a 0/1 loss function (Equation 2.9) on segment sets. Under these conditions the segmental MBR recognizer of Equation 2.30 becomes

$$\delta(A^i) = \underset{W'^i \in \mathcal{W}_h^i}{\operatorname{argmax}} P^i(W'^i|A), \tag{2.33}$$

where $P^i(W'^i|A)$ is defined in a manner similar to that of Equation 2.11.
Equation 2.33 is none other than the maximum a-posteriori probability decision on each hypothesis segment set: for each hypothesis word a marginal probability is computed based on the evidence space. The word with highest marginal probability is then selected. This is the procedure of segmental voting.
The utterance level induced loss (Equation 2.32) for segmental voting can be written as

$$l_{\text{seg-vote}}(W, W') = \sum_{i=1}^{N} l_{0/1}(R_e^i(W), R_h^i(W')). \tag{2.34}$$

As is the case with segmental MBR recognition, segmental voting is effective if $l_{seg-vote}$ is a good approximation to the loss that we are trying to minimize.
Segmental MBR recognition does not specify how to find the hypothesis and evidence segment set segmentation procedures R_h and R_e; it only specifies the constraints that these procedures must obey. The construction of segment sets therefore remains a design problem to be addressed in an application specific manner. We will now describe two versions of segmental MBR recognition used in state-of-the-art

ASR systems. Both these procedures attempt to reduce the word error rate (WER) and thus are based on the Levenshtein loss function [26].

2.3.2 ROVER

Recognizer output voting for error reduction (ROVER) is an N-best list segmental voting procedure. It combines the hypotheses from multiple independent recognizers under the Levenshtein loss. In its original formulation [5], each of these outputs consists of a single word string and a word level confidence score associated with each word in that string. Procedures for combining N-bests lists from each system have since been developed [4, 11].

Let $\mathcal{N}_m, m = 1, ..., K$ be N-best lists produced by K recognition systems in response to acoustics A, and let P_m be the posterior distribution associated with $\mathcal{N}_m$. Let $\mathcal{N}_e$ denote the union of these N-best lists. A posterior distribution on word strings in $\mathcal{N}_e$ is derived by first extending each P_m to assign zero probability to word strings in $\mathcal{N}_e$ that are not present in $\mathcal{N}_m$ and then taking a convex combination

$$P(W|A) = \sum_{m=1}^{K} \alpha_m P_m(W|A), \quad W \in \mathcal{N}_e, \quad \sum_{m=1}^{K} \alpha_m = 1. \tag{2.35}$$

The set $\mathcal{N}_e$ and $P(W|A)$ are the evidence space and the evidence distribution used by ROVER. †

The word strings of $\mathcal{N}_e$ are arranged in a word transition network (WTN) that represents an approximate *simultaneous alignment* of these hypotheses. It is generated by picking top two hypotheses, aligning them to produce an initial WTN, and then iteratively adding each new hypothesis by aligning it with the WTN constructed so far. An example WTN produced by aligning:

{"OH WELL WE," "O WELL WE'RE," "WELL WE WE'RE"}

is given in Figure 2.2. A set of words that align with each other is called a *correspondence set*.

The WTN trivially specifies an evidence segmentation rule R_e for word strings of $\mathcal{N}_e$. The hypothesis space of ROVER is the set of all the word strings that can be produced by picking one word from each correspondence set and concatenating them. Therefore, the hypothesis segmentation rule R_h and the conjunction rule J_h are also trivially specified by the WTN. Having segmented the evidence and hypothesis spaces, a marginal probability is computed for each word in each correspondence set according to Equation 2.31 and the word with the largest marginal probability is chosen from each correspondence set. These words are concatenated to form the final output of ROVER.

† ROVER originally incorporated a word level confidence score instead of $P(W|A)$ as in Equation 2.35; this is discussed by Goel et al. [9].

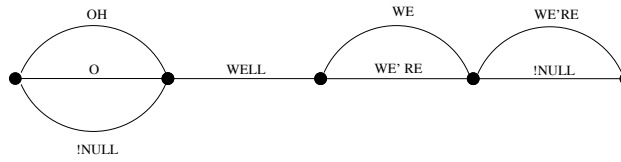

FIGURE 2.2
An example word transition network.

The utterance level induced loss (Equation 2.32) in ROVER is derived from Equation 2.34 where the sum is over the correspondence sets

$$l_{ROVER}(W, W') = \sum_{i=1}^{N} l_{0/1}(R_e^i(W), R_h^i(W')). \tag{2.36}$$

This loss is similar to the Levenshtein distance between strings W and W' when their alignment is specified by the WTN. Since the WTN construction process adds each new word string to the WTN so as to minimize the alignment cost between that string and the WTN, we could expect $l_{ROVER}(W, W')$ to approximate the Levenshtein distance between W and W'.

2.3.3 e-ROVER

The simultaneous alignment produced in ROVER may sometimes be suboptimal for some evidence hypotheses pairs. The natural remedy is to allow multiple consecutive words in each correspondence set. Considering the search for segment sets as a clustering problem, there are two different approaches that could be taken. We could take a 'top-down' approach that starts with a single correspondence set that contains an entire N-best list and segments it into sets that contain shorter word strings. Alternatively, we could take a 'bottom-up' approach where we first construct a WTN that contains no more than one consecutive word in each correspondence set. We could then join consecutive sets to obtain sets with longer word strings. The procedure of extended-ROVER (e-ROVER) starts with the ROVER WTN and takes the latter, bottom-up, approach.

The process of *joining* two correspondence sets yields one *expanded* set that contains all the paths from the original pair of correspondence sets. This is graphically illustrated in Figure 2.3.

The utterance level loss function of e-ROVER is given as follows. Starting from the initial WTN, let two consecutive correspondence sets, say sets m and $m + 1$, be joined, and let the loss function on the expanded set be the Levenshtein distance. The loss function on correspondence sets that did not expand remains the 0/1 loss.

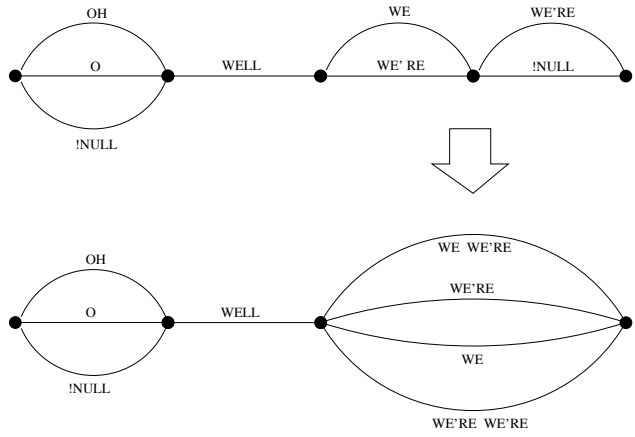

FIGURE 2.3
Joining two correspondence sets.

The utterance level loss is then

$$l_{e-ROVER}(W, W') = \sum_{i=1, i\neq m, i\neq m+1}^{N} l_{0/1}(R_e^i(W), R_h^i(W')) + L(W^m, W'^m). \tag{2.37}$$

Here, W^m and W'^m are word subsequences from the joined segment sets. It follows from the definition of Levenshtein distance that

$$L(W, W') \leq l_{e-ROVER}(W, W') \leq l_{ROVER}(W, W').$$

This follows because the e-ROVER alignments will eventually achieve the Levenshtein alignment as the alignment constraints are reduced.
The joining procedure can be carried out many times to yield successively better approximations to the Levenshtein distance. The WTN obtained after each joining operation specifies a new segmentation of the evidence and hypothesis spaces. In comparing e-ROVER to ROVER, it is important to note that only the *segmentation* of the hypothesis and evidence spaces changes with joining operation, the actual spaces remain the same as they were in ROVER.
There are two consequences of joining correspondence sets. First, after the joining operation, the loss function on the expanded set is no longer the 0/1 loss but is instead the Levenshtein distance. Hence the MBR hypothesis selection on this set needs to follow Equation 2.30. Second, the size of the expanded set grows exponentially with the number of joining operations, making Equation 2.30 progressively difficult to implement. It is therefore important to determine the sets to be joined carefully so as to yield maximum gain in Levenshtein distance approximation with minimum combinations of the correspondence sets. A heuristic procedure for joining sets [9] is based on first identifying correspondence sets in which the largest value of the

marginal probability (Equation 2.31) is below a threshold. Each consecutive stretch of such sets is joined to form one expanded set. Sets in which the largest value of the marginal probability is above the threshold are kept 'pinched'; they are not joined with any other set. For details of this procedure readers are referred to Goel et al. [9]. As noted above, the hypothesis and the evidence spaces in e-ROVER are identical to those in ROVER. However, the loss function in e-ROVER provides a better approximation to the word error rate than ROVER. Since they are both instantiations of Equation 2.29, e-ROVER would be expected, in theory, to yield a lower word error rate than ROVER.

2.4 Experimental Results

The minimum Bayes-risk procedures yield a theoretically lower expected error rate than the MAP recognizer. However, their practical merit can only be gauged in real classification tasks. In this section we present experiments that compare MBR and segmental MBR procedures with MAP recognition and with each other.
We first address a practical problem associated with incorporation of the HMM and Markov chain models into the minimum-risk search procedures described above.

2.4.1 Parameter Tuning within the MBR Classification Rule

The joint distribution $P(W, A)$ to be used in the MBR recognizers is derived by combining probabilities from acoustic and language models. It is often found useful in practice to introduce some tuning parameters to help match these models better. In the following we discuss a parameterization of $P(W, A)$ that is suitable for use in MBR recognizers. We then present strategies to optimize these parameters within the MBR classification rule.
It is customary in ASR to use two tuning parameters in the computation of joint probability

$$P_{\alpha,\beta}(W, A) = e^{\alpha|W|}P(A|W)P(W)^{\beta} \tag{2.38}$$

where $|W|$ is the number of words in word string W. The parameter α, usually a negative constant, causes a decrease of probability with increasing $|W|$. For this reason it is called *word insertion penalty*. The other parameter β scales the language model probability relative to the acoustic model probability; it is termed *language model scale factor*.
We have found it useful to introduce an additional *likelihood scale factor* γ [14]

$$P_{\alpha,\beta,\gamma}(W, A) = \{e^{\alpha|W|}P(A|W)P(W)^{\beta}\}^{1/\gamma}. \tag{2.39}$$

The likelihood scale factor restricts the dynamic range of the probabilities. For example, consider the 10-best list of Table 2.1. These are ten most likely word strings pro-

TABLE 2.1
Example ten most likely hypotheses and the posterior probability of these hypotheses under two different parameterizations (Equations 2.38 and 2.39) of the posterior distribution. $\gamma = 15.0$, $\alpha = -10.0$, and $\beta = 12.0$.

$\ln P_{\alpha,\beta}(W, A)$	$P_{\alpha,\beta}(W\|A)$	$P_{\alpha,\beta,\gamma}(W\|A)$	Sentence
-22402.56	1.0	0.3547	I HAVE A RURAL AREA
-22420.05	2.5E-08	0.1077	I HAVE A REAL RURAL AREA
-22420.10	2.4E-08	0.1074	ALTHOUGH IN A RURAL AREA
-22422.69	1.8E-09	0.0903	I LIVE IN A RURAL AREA
-22425.15	1.5E-10	0.0767	ALTHOUGH IT WILL AREA
-22428.33	6.4E-12	0.0620	SO I HAVE A RURAL AREA
-22428.35	6.3E-12	0.0619	HAVE A RURAL AREA
-22430.66	6.2E-13	0.0531	I'M A RURAL AREA
-22431.63	2.4E-13	0.0498	I HAVE A LITTLE RURAL AREA
-22433.05	5.7E-14	0.0453	I HAVE A ROLE AREA

duced by our ASR system for an utterance "I LIVE IN A RURAL AREA". The log-likelihoods ($\ln_{\alpha,\beta} P(W, A)$) along with the posterior distributions computed with and without the likelihood scale factor are shown in Table 2.1. The posterior distribution $P_{\alpha,\beta}(W|A)$ is computed by exponentiating the log-likelihoods and then normalizing them over the ten-best list.

From Table 2.1 it can be seen that $P_{\alpha,\beta}(W|A)$ is heavily weighted towards the most likely candidate, owing to a large variation in log-likelihood values. This leads to a degeneration of the evidence space. It is prevented by the introduction of the likelihood scale factor which flattens the distribution and yields more reasonable posterior probabilities $P_{\alpha,\beta,\gamma}(W|A)$. A value of $\gamma = 15.0$ is used in our example of Table 2.1.

2.4.1.1 Optimization of Likelihood Parameters

Let $\delta_{\alpha,\beta,\gamma}$ be the minimum-risk recognizer (Equation 2.4) incorporating the parameterized distribution $P_{\alpha,\beta,\gamma}(W, A)$ of Equation 2.39. We optimize α, β, and γ to minimize the empirical risk [37] of $\delta_{\alpha,\beta,\gamma}$,

$$\sum_{(W,A)\in\mathcal{T}} l(W, \delta_{\alpha,\beta,\gamma}(A)), \tag{2.40}$$

over a database $\mathcal{T} = \{(W, A)\}$ of labeled utterances. Since the utterance labels are known, this is *supervised optimization*.

For some problems it may be desirable to tune classification rule parameters without using a separate training set. We approach this *unsupervised optimization* problem by minimizing the empirical risk (Equation 2.40) using the most likely evidence string in place of the truth. This evidence string is removed from the evidence space; otherwise the empirical risk would be minimized by placing a probability mass of 1.0 on this evidence string by the degenerate parameter value of $\gamma \to 0.0$. Furthermore, to reduce the bias of unsupervised training towards the most likely evidence string, we remove all those hypotheses that are at zero loss from this evidence string as well as all those evidence strings that are at zero loss from any of the hypotheses removed.

In order to reduce the number of parameters to be trained, we kept the word insertion penalty (α) and language model scale factor (β) fixed at their values obtained from training with MAP classifier. In all our experiments reported in this thesis these values were $\alpha = -10.0$ and $\beta = 12.0$. A grid search for optimal γ was performed in both supervised and unsupervised optimization.
An alternative to training γ is to use $\gamma = \beta$ [36]. We compare all three methods for obtaining γ described here in the experiments to follow.

2.4.2 Utterance Level MBR Word and Keyword Recognition

We now evaluate utterance level N-best list rescoring (Section 2.2.2) and prefix tree based A^* search (Section 2.2.3.4) for tasks of transcription and keyword spotting.
Transcription is the task of identifying word content of spoken acoustics. Its error rate is measured by the Levenshtein distance, $L(W, W')$, between the actually spoken utterance and the recognizer's output.
The goal of keyword spotting is to identify the presence, and sometimes the time location, of a prespecified set of keywords. A loss function suitable for such a task would pay attention only to the keywords; other spoken words should be ignored. We chose to experiment with a loss function based on a variant of Levenshtein distance that assigns a cost of one when there is an error on a keyword, and assigns no cost to errors on other words. The definition used was

$$l(W, W') = L(W_c, W'_c) \tag{2.41}$$

where $L(\cdot, \cdot)$ is the Levenshtein distance, and W_c is derived from W by deleting all its non-keywords. ASR performance measured under $L(W_c, W'_c)$ will be referred to as keyword error rate (KER).
Experiments were conducted on the Switchboard [7] corpus that consists of spontaneous telephony conversations between individuals. The test set was a linguistically segmented subset of this corpus, used for the 1997 Johns Hopkins University LVCSR Workshop [21]. This test set contained 2427 utterances from 38 conversation sides; the complete test set definition and other details can be found in the workshop proceedings. Word lattices were generated under a trigram language model using speaker and gender independent, HTK-based, 12-component Gaussian mixture cross-word triphone system [40] with 6973 triphone states. For use in the N-best list rescoring procedure of Equation 2.16, 1000 element N-best lists were generated from the trigram lattices. These were used as the evidence space $\mathcal{N}_e$ and from them the top 25 elements were kept as the hypothesis space $\mathcal{N}_h$. The MAP candidate in these N-best lists (and hence in these lattices) served as the baseline with a word error rate of 38.5% and a sentence error rate (SER) of 65.9%.
Words in the task vocabulary were marked as keywords if they occurred relatively infrequently in a large corpus [6]. Examples of the two kinds of words are

Keywords (21,653)	: *abilities, bartenders, calculation, databases*
Non-keywords (784)	: *a, and, the, besides, collaboration, distribution*

The numbers in parentheses above denote the total number of distinct words of that kind in the system vocabulary of size 22,437. Even though the non-keywords constitute a small fraction of the vocabulary, they are quite abundant and account for more than 70% of the word tokens. The full vocabulary with marked keywords can be found at our web site [8].

2.4.2.1 Likelihood Scale Factor Tuning

The likelihood scale factor was tuned as described in Section 2.4.1. Supervised optimization used a held out data set of 2218 utterances from 1040 conversation sides that was separate from the training or test sets. Unsupervised optimization was performed on the test set itself. Each entire 1000 element N-best list was used as the evidence space $\mathcal{N}_e$, and the top 25 elements were kept as the hypothesis space $\mathcal{N}_h$ for parameter tuning. For unsupervised optimization under KER, we removed the MAP candidate from the N-best list. We also removed all the N-best entries that had zero KER with respect to the MAP candidate. This was done for the reasons described in Section 2.4.1. Parameter tuning was also compared with the alternative approach of using the language model scale factor (Section 2.4.1); these comparisons are presented in Table 2.2.

2.4.2.2 N-best List Rescoring and A^* Search

The N-best list rescoring procedure of Section 2.2.2 was implemented with a 1000 element evidence space and a 25 element hypothesis space for both WER and KER. Results of this rescoring under WER are listed in Section A of Table 2.2 and those under KER are listed in Section B of Table 2.2. Looking under the WER column in Section A and under the KER column in Section B, we note that N-best list rescoring yields a small yet significant improvement over corresponding MAP baselines. Furthermore, rescoring for WER is not affected by the likelihood scale factor selection method, whereas for KER the unsupervised optimization method outperforms the other two methods.

The multistack prefix tree based procedure described in Section 2.2.3.4 was implemented for A^* search. The extension to keyword spotting is straightforward since the task loss function is based on the Levenshtein distance. Two forms of pruning were used during the A^* search. For each partial hypothesis, its MAP completion was found. The partial hypothesis was discarded if this probability fell below a threshold set with respect to the MAP lattice hypothesis. Partial hypotheses were also pruned by their cost under estimates (Equation 2.24). Under these two pruning conditions the prefix tree search took approximately twice as long as the N-best list rescoring procedure.

Looking at the WER performance of WER optimized search and KER performance of KER optimized search in Table 2.2, we note that the A^* search yields significant error rate reduction over the corresponding N-best list rescoring procedures. The importance of unsupervised optimization method of likelihood scale parameter is also more prominent in this case.

An overall increase in the WER for hypotheses optimized for KER, and the KER

TABLE 2.2
Evaluation of parameter tuning and recognition procedures for minimization of WER and KER.

Baseline (MAP): WER = 38.5, KER = 43.2							
	Recognition & Tuning Criterion	Parameter Tuning Strategy	Likelihood Scale Factor (γ)	Recognition Strategy			
				N-best		A^*	
				WER	KER	WER	KER
A	WER	LM Scale	12.0	37.9	42.9	37.7	42.5
	WER	Supervised	15.2	37.9	43.0	37.5	42.4
	WER	Unsupervised	15.2	37.9	43.0	37.5	42.4
B	KER	LM Scale	12.0	38.7	42.5	N/A	41.9
	KER	Supervised	18.0	38.7	42.4	N/A	41.6
	KER	Unsupervised	15.5	38.8	42.0	N/A	41.4

performance of hypotheses optimized for WER reinforces that, as desired, a task specific minimum-risk classifier outperforms classifiers optimized for other tasks.

2.4.3 ROVER and e-ROVER for Multilingual ASR

In this section we evaluate the N-best list based segmental MBR procedures of ROVER (Section 2.3.2) and e-ROVER (Section 2.3.3). We will apply these methods to multilingual, language independent acoustic modeling [2]. The objective here is to train a monolingual system on a small amount of transcribed speech and then to improve its performance using acoustic model trained in other languages. One of the claimed advantages of ROVER techniques is the ability to combine multiple ASR systems to generate a single hypothesis. We will show that ROVER does indeed improve over the performance of a monolingual system, and that e-ROVER can be used for further improvements.

Three systems were combined: a triphone system trained on one hour of Czech voice of America (CZ-VOA) database[‡] (Sys1); a triphone system trained on 72 hrs. of English and then adapted by one hour of Czech voice (Sys2); and Sys1 output rescored with Sys2 models (Sys3). The test set consisted of 748 held out utterances from CZ-VOA broadcast.

ASR lattices were generated using the one hour Czech voice based monolingual system. By rescoring these lattices a set of 250 hypotheses was generated for each system. The MAP hypotheses (top candidate in the N-best lists) in these three systems had error rates of 29.58%, 35.24%, and 29.22%, respectively. We note that the performance of the English system (Sys2) was substantially worse when not constrained by the first-pass lattices produced by the Czech monolingual system.

For each system, the word insertion penalty and the language model scale factor (Section 2.4.1) were chosen to yield optimal performance by the MAP decision

[‡] Available from the Linguistic Data Consortium, LDC2000S89 Voice of America (VOA) Czech Broadcast News Audio.

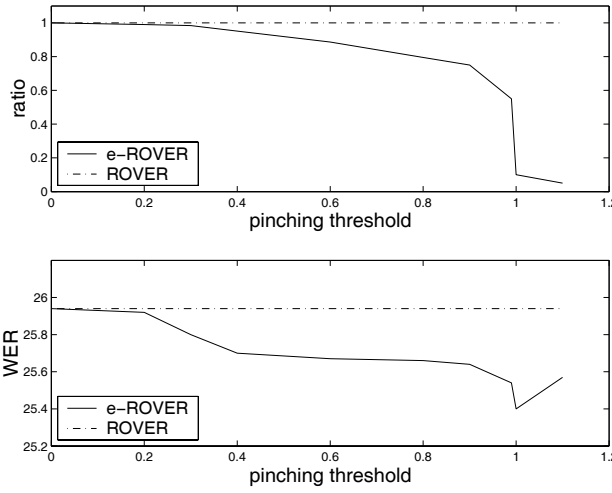

FIGURE 2.4
Top panel shows the ratio of total number of e-ROVER correspondence sets to that of ROVER correspondence sets, as a function of the pinching threshold. Bottom panel shows the WER performance of e-ROVER for these thresholds.

rule. The likelihood scale factor was obtained by conducing an unsupervised optimization (Section 2.4.1) separately for each system. ROVER and e-ROVER were implemented by combining these three sets of 250 hypotheses. The posterior distribution over the resulting 750-best list was derived by simply renormalizing the log-likelihoods of the (scaled) individual hypotheses.

2.4.3.1 Correspondence Set Pinching

In e-ROVER the correspondence sets were joined using the heuristic procedure described in Section 2.3.3. This procedure joins the correspondence sets based on a "pinching threshold" that considers the largest posterior probability of any word string in each correspondence set. A threshold of 0.0 results in no joining at all - which is equivalent to ROVER, while any threshold above 1.0 merges all the correspondence sets.

Our implementation of ROVER resulted in a 25.94% WER, which is a 3.28% absolute improvement over the best MAP word error rate of the three systems being combined. Figure 2.4 shows that additional gains can be obtained using e-ROVER. The top panel shows the ratio of total number of e-ROVER correspondence sets to total number of ROVER correspondence sets as a function of the pinching threshold. This ratio is 1.0 for threshold value of 0.0, and decreases monotonically as the threshold increases. It is not at its minimum for a threshold of 1.0 due to the presence of correspondence sets which contain only one word; these sets have a word with marginal probability of 1.0 and remained pinched for a threshold value of 1.0.

The bottom panel in Figure 2.4 shows the effect of pinching on WER. We note that all thresholds result in better than ROVER word error rate. The threshold of 1.0 yields the best performance of 0.56% absolute improvement over ROVER and hence a total of 3.84% absolute over the best baseline error rate. We see a degradation in performance for thresholds larger than 1.0. One possible explanation is the need for heavier pruning due to the greatly enlarged search space that results from expanding all the segment sets. Another possibility is that the best strategy is to retain the word segments that were recognized with absolute certainty by the first-pass system.

2.5 Summary

We have described automatic speech recognition algorithms that attempt to minimize the average misrecognition cost under task specific loss functions. These recognizers, although generally more computationally complex than more widely used MAP algorithms, can be efficiently implemented using an N-best list rescoring procedure or as an A^* search over recognition lattices. While the A^* is generally more accurate, its implementation requires that upper and lower bounds on the cost of partial hypotheses be computed as the search proceeds. These must be derived for each performance criterion of interest, and we have given expressions for the Levenshtein and keyword error rates. In LVCSR experiments we have shown that MBR decoding procedures can be used to tune ASR performance for task specific loss functions.

Segmental MBR is described as a special case of MBR recognition that results from the segmentation of the recognition search space. The segmentation is done with the assumption that the loss function induced is a good approximation to the original, desired loss function. It is discussed how recognizer voting can be considered in the SMBR framework, and in particular, the widely-used ROVER system combination procedure is described in this way. That ROVER can be described as an MBR procedure under a loss function related to the WER provides a plausible explanation for the performance improvements that it has been found to provide. We then described e-ROVER, which is a ROVER variant based on a loss function that can be tuned to better approximate the Levenshtein distance. The value of these technique are demonstrated by using ROVER and e-ROVER for multilingual system combination. As has been shown in these and other experiments, recognizer voting procedures can combine recognition hypotheses from diverse systems to generate a single hypothesis that is better than the best hypothesis of any of the individual systems. These experiments were based on the segmentation of N-best lists produced by each system. However similar procedures can be derived for lattice rescoring, and the development of MBR lattice segmentation procedures is a topic of current research.

2.6 Acknowledgements

We thank Dimitra Vergyri for providing the lattices that were used in our experiments and Kumar Shankar for help with the experiments. We also thank Andreas Stolcke and Lidia Mangu for useful discussions.

References

[1] P. J. Bickel and K. A. Doksum. *Mathematical Statistics: Basic Ideas and Selected topics*. Holden-Day Inc., Oakland, CA, 1977.

[2] W. Byrne, P. Beyerlein, J. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Peterek, J. Picone, D. Vergyri, and W. Wang. Towards language independent acoustic modeling. In *IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 1029–1032, Istanbul, Turkey, 2000.

[3] N. Chinchor, P. Robinson, and E. Brown. Hub-4 Named Entity Task Definition Version 4.8. In 9^{th} *Hub-5 Conversational Speech Recognition Workshop*, 1998. Available at www.nist.gov/speech/hub4_98.

[4] G. Evermann and P. Woodland. Posterior Probability Decoding, Confidence Estimation and System Combination. In *In Proceedings of the NIST and NSA Speech Transcription Workshop*, College Park, MD, 2000.

[5] J. Fiscus. A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER). In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–354, 1997.

[6] Radu Florian and David Yarowsky. Dynamic Nonlocal Language Modeling via Hierarchical Topic-Based Adaptation. In *ACL99*, pages 167–174, 1999.

[7] J. J. Godfrey, E. C. Holliman, and J. McDaniel. Switchboard: Telephone Speech Corpus for Research and Development. In *IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520, San Francisco, CA, 1992.

[8] V. Goel. *Word List With Content Word Marks*, 1999. Available at http://www.clsp.jhu.edu/people/vgoel.

[9] V. Goel. *Minimum Bayes-Risk Automatic Speech Recognition*. Ph.D. Dissertation, Johns Hopkins University, Baltimore, MD, 2001.

[10] V. Goel and W. Byrne. Task Dependent Loss Functions in Speech Recognition: A^* Search over Recognition Lattices. In *Eurospeech-99*, pages 1243–1246, Budapest, Hungary, 1999.

[11] V. Goel and W. Byrne. Applications of Minimum Bayes-Risk Decoding to LVCSR. In *In Proceedings of the NIST and NSA Speech Transcription Workshop*, College Park, MD, 2000.

[12] V. Goel and W. Byrne. Minimum Bayes-Risk Automatic Speech Recognition. *Computer Speech and Language*, 14(2):115–135, 2000.

[13] V. Goel and W. Byrne. Recognizer Output Voting and DMC in Minimum Bayes-Risk Framework. In *Research Notes No. 40, Center for Language and Speech Processing*, 2000.

[14] V. Goel, W. Byrne, and S. Khudanpur. LVCSR Rescoring With Modified Loss Functions: A Decision Theoretic Perspective. In *IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 425–428, 1998.

[15] V. Goel, S. Kumar, and W. Byrne. Segmental Minimum Bayes-Risk ASR Voting Strategies. In 6^{th} *International Conference on Spoken Language Processing*, volume 3, pages 139–142, Beijing, China, 2000.

[16] P. S. Gopalakrishnan, L. R. Bahl, and R. L. Mercer. A Tree Search Strategy for Large Vocabulary Continuous Speech Recognition. In *IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 572–575, 1995.

[17] P. E. Hart, N. J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4(2):100–107, 1968.

[18] P. E. Hart, N. J. Nilsson, and B. Raphael. Correction to 'A Formal Basis for the Heuristic Determination of minimum Cost Paths'. *SIGART Newsletter*, 37:28–29, 1972.

[19] F. Jelinek. A Fast Sequential Decoding Algorithm Using a Stack. *IBM Journal of Research Development*, 13:675–685, 1969.

[20] F. Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts, 1997.

[21] *Proceedings of the 1997 Large Vocabulary Continuous Speech Recognition Workshop*, 1997. Available at http://www.clsp.jhu.edu/ws97.

[22] B-H. Juang and S. Katagiri. Discriminative Learning for Minimum Error Classification. *IEEE Transactions on Signal Processing*, SP-40(12):3043–3054, 1992.

[23] J. Kaiser, B. Horvat, and Z. Kacic. A Novel Loss Function for the Overall Risk Criterion Based Discriminative Training of HMM Models. In 6^{th} *International Conference on Spoken Language Processing*, volume 2, pages 887–890, Beijing, China, 2000.

[24] T. Kawahara, C-H. Lee, and B-H. Juang. Combining Key Phrase Detection and Subword Based Verification for Flexible Speech Understanding. In *IEEE*

Conference on Acoustics, Speech, and Signal Processing, volume 1, pages 193–196, 1997.

[25] M. W. Koo, C-H. Lee, and B-H. Juang. A New Hybrid Decoding Algorithm for Speech Recognition and Utterance Verification. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 303–310, 1997.

[26] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Phys. Dokl.*, 10:707–710, 1966.

[27] L. Mangu, E. Brill, and A. Stolcke. Finding Consensus Among Words: Lattice-Based Word Error Minimization. In *Eurospeech-99*, pages 495–498, Budapest, Hungary, 1999.

[28] A. Martin, J. Fiscus, M. Przybocki, and B. Fisher. 1998 Hub-5 Workshop: Information Retrieval. In *9th Hub-5 Conversational Speech Recognition Workshop*, 1998.

[29] A. Martin, J. Fiscus, M. Przybocki, and B. Fisher. 1998 Hub-5 Workshop: Weighted Word Results. In *9th Hub-5 Conversational Speech Recognition Workshop*, 1998.

[30] K. Na, B. Jeon, D. Chang, S. Chae, and S. Ann. Discriminative Training of Hidden Markov Models Using Overall Risk Criterion and Reduced Gradient Method. In *Eurospeech-95*, pages 97–100, Madrid, Spain, 1995.

[31] A. Nadas. A Decision Theoretic Formulation of the Training Problem in Speech Recognition and a Comparison of Training by Unconditional Versus Conditional Maximum Likelihood. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31(4):814–817, 1983.

[32] A. Nadas. Optimal Solution of a Training Problem in Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-33(1):326–329, 1985.

[33] D. B. Paul. An Efficient A^* Stack Decoder Algorithm for Continuous Speech Recognition with a Stochastic Language Model. In *IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 25–28, 1992.

[34] E. Ristad and P. Yianilos. Learning String Edit Distance. *IEEE Trans. PAMI*, 20(5):522–532, 1998.

[35] R. C. Rose and D. B. Paul. A Hidden Markov Model Based Keyword Recognition System. In *IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 129–132, 1990.

[36] A. Stolcke, Y. Konig, and M. Weintraub. Explicit Word Error Minimization in N-Best List Rescoring. In *Eurospeech-97*, volume 1, pages 163–165, Rhodes, Greece, 1997.

[37] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.

[38] F. Wessel, R. Schluter, and H. Ney. Using Posterior Word Probabilities For Improved Speech Recognition. In *IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1587–1590, Istanbul, Turkey, 2000.

[39] J. G. Wilpon, L. R. Rabiner, C-H. Lee, and E. R. Goldman. Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-38(11):1870–1878, 1990.

[40] S. Young. *HTK 2.1*. Entropic Cambridge Research Laboratory Ltd., Cambridge, UK, 1997.

3

A Decision Theoretic Formulation for Robust Automatic Speech Recognition

Qiang Huo
The University of Hong Kong, Hong Kong, China

CONTENTS

3.1 Introduction

Modern automatic speech recognition (ASR) technology [9, 10, 56, 105, 84, 57, 20, 125, 46] is based on a communication theoretic view of the generation, acquisition and transmission, and perception of speech [6]. Figure 3.1 (adapted from Juang's keynote speech in NNSP'96 [68]) shows a conceptual model of a noisy channel for speech generation and signal capturing. The goal of speech recognition is then defined as recovering the word sequence, W, from the acoustic signal, $\mathbf{X}$. This can also be viewed as a *decision problem*, i.e., based on the information in $\mathbf{X}$ and the other relevant aspects of the problem, we attempt to make the best decision (in some sense) of the W that has been embedded in $\mathbf{X}$. For the simplicity of discussion, we can view each possible word sequence W as a *class*. Let us assume there are total M unique classes. So, speech recognition consists in finding optimal (in some sense) decision rules for classification of the observation $\mathbf{X}$ into one of M fixed classes. Depending on different criteria, there exist many decision rules. Not all of them are of equal value in practice. Because of the different sources of variability as shown in Figure 3.1, the speech signal $\mathbf{X}$ is usually featured by uncertainty, variability, lack of determinism, and stochasticity. This makes the *statistical pattern recognition* approach [100, 44, 18, 71, 55, 19] a natural choice for formulating and solving the

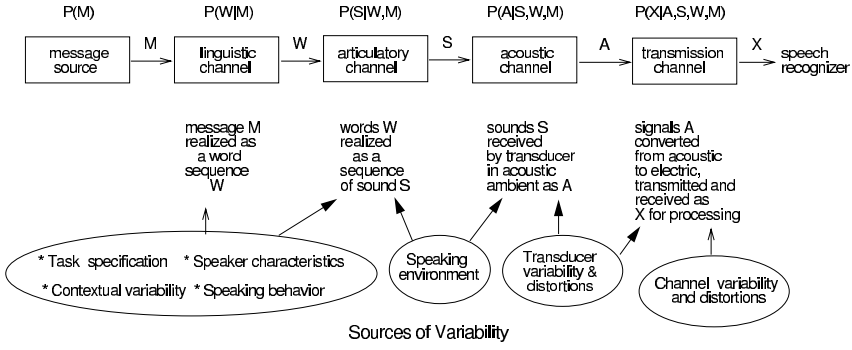

FIGURE 3.1
Communication Theoretic View of ASR: Noisy Channel for Speech Generation and Signal Capturing (adapted from [68]).

ASR problem as described briefly in the following.
First, the statistical models for the channels in Figure 3.1 are simplified as follows:

- A word sequence W and the associated acoustic observation $\mathbf{X}$ are viewed as a jointly distributed random pair $(W, \mathbf{X})$. For notational simplicity, we will use the same symbol to denote both the random variable and the value it may assume.

- The joint distribution of $(W, \mathbf{X})$ is modeled by a *parametric family* of PDF (probability density function) $p(W, \mathbf{X}) = p_\Lambda(\mathbf{X}|W) \cdot P_\Gamma(W)$; $p_\Lambda(\mathbf{X}|W)$ is known as the acoustic model with parameters Λ, and $P_\Gamma(W)$ as the language model with parameters Γ.

- The parameters (Λ, Γ) of the above distributions are to be estimated from some *training data* by using particular parameter estimation techniques.

With these simplifications, the most popular way to solve the ASR problem is to use the well-known *plug-in MAP* (maximum *a posteriori*) *decision rule* [18, 106, 79],

$$\hat{W} = \operatorname*{argmax}_W P(W|\mathbf{X}) = \operatorname*{argmax}_W p_{\hat{\Lambda}}(\mathbf{X}|W) \cdot P_{\hat{\Gamma}}(W) \,, \tag{3.1}$$

where $\hat{\Lambda}$ and $\hat{\Gamma}$ are the estimated parameters obtained during training, and $\hat{W}$ is the recognized sentence during testing. This decision rule, derived from the optimal Bayes' decision rule, is also widely used in many other pattern recognition applications.

This chapter attempts to explain, from a *statistical decision* point of view, why the above approach works so well in certain conditions, and more importantly why it does not work in many other situations. To do this, Section 3.2 first explains the decision theoretic formulation of the ASR problem and the optimal decision rule that can be constructed if everything about the problem is known. Then Section 3.3 explains how to construct the adaptive decision rules when learning from a training sample set. The rationale of two popular design principles in constructing such adaptive decision rules is also discussed. Section 3.4 discusses the classification of possible distortions of hypothetical models and data, and the possible ways of achieving performance robustness. Section 3.5 reviews and discusses some of the recent parameter adaptation techniques for improving adaptive decision rules. Section 3.6 explains the basic notation of the decision rule robustness and shows two examples of how to construct robust decision rules, namely the minimax decision rule and the Bayesian predictive classification rule. Section 3.7 summarizes the ideas discussed in the chapter.

3.2 Optimal Bayes' Decision Rule for ASR

In its simplest form, let us assume that our ASR problem is to classify a speech observation (in practice, usually a feature vector sequence extracted from the speech signal) $\mathbf{X}$ into one of M classes, $W \in \Omega_W$, where $\Omega_W = \{W_1, W_2, \cdots, W_M\}$ denotes the set of M classes. Depending on the problem of interest, a class $W \in \Omega_W$ may be of any linguistic unit, e.g., a phoneme, a syllable, a word, a phrase, a sentence, a semantic concept or attribute, etc. Let us assume that the speech observation $\mathbf{X}$ belongs to a suitable space Ω_x. The problem of constructing a speech recognizer is then equivalent to finding a *decision rule* $d(\cdot)$ in a set of possible decision rules $\mathcal{D}$, such that $d : \Omega_x \rightarrow \Omega_W$, or simply

$$W = d(\mathbf{X}), \quad \text{for } \mathbf{X} \in \Omega_x, W \in \Omega_W, \text{ and } d(\cdot) \in \mathcal{D}, \tag{3.2}$$

with W being one of the M possible class labels in Ω_W. In this case, the *decision space*, $\Omega_D = \{d(\mathbf{X}) : \mathbf{X} \in \Omega_x\}$, of the decision rule $d(\cdot)$ is the same as the Ω_W. A decision rule $d(\cdot) \in \mathcal{D}$ implies a mapping from the sample space to the class label space. This mapping is known as a *nonrandomized decision rule* [24]. Define $\Omega_x(W_i) = \{\mathbf{X} : \mathbf{X} \in \Omega_x, d(\mathbf{X}) = W_i\}$ to be a subset of Ω_x corresponding to the region of $\mathbf{X}$ being mapped as class W_i with the decision rule $d(\cdot)$, then the construction of a decision rule amounts to finding a partition, $\Omega_x(d(\cdot)) = \{\Omega_x(W_1), \Omega_x(W_2), \ldots, \Omega_x(W_M)\}$, of the observation space Ω_x under the following constraints:

$$\bigcup_{i=1}^{M} \Omega_x(W_i) = \Omega_x, \quad \Omega_x(W_i) \bigcap \Omega_x(W_j) = \emptyset, \quad \text{for } i \neq j; \quad i, j = 1, 2, \cdots, M.$$

There may exist an infinite set of decision rules for the same given classification problem. Not all of them are of equal value in practice though. To determine whether a decision rule is "good" one has to agree on a reasonable set of criteria for assessing the "goodness". Let us show one possible formulation by using the classical statistical decision theory pioneered by Wald and developed by many others [121, 24].
Let us view W and an observation $\mathbf{X}$ as a jointly distributed random pair $(W, \mathbf{X})$, whose joint PDF is denoted by $p(W, \mathbf{X})$. In the so-called *sampling paradigm*, we can decompose $p(W, \mathbf{X})$ into a product of the class prior probability $P(W)$ and the class conditional PDF $p(\mathbf{X}|W)$, i.e., $p(W, \mathbf{X}) = p(\mathbf{X}|W)P(W)$. One way of formalizing a goodness criterion is to use the knowledge of the possible consequences of the decisions. Often this knowledge can be quantified by assigning a *loss* that would be incurred for each possible decision. Let $\ell(W, d(\mathbf{X}))$ be the *loss function* associated with making a decision, $d(\mathbf{X})$, if the true class is W. One would like the loss function to have the following property

$$0 \leq \ell(W, W) \leq \ell(W, d(\mathbf{X}) \neq W). \tag{3.3}$$

If we assume the *true distribution* $p(W, \mathbf{X})$ is known, then the conditional and marginal distributions, namely, $p(\mathbf{X}|W)$, $p(W|\mathbf{X})$, $P(W)$, and $p(\mathbf{X})$ can be calculated. Now we can define the *total risk*, $r(d(\cdot))$, for a decision rule $d(\cdot)$ as an expected value of the loss function, i.e.,

$$\begin{aligned} r(d(\cdot)) &= \mathbf{E}_{(W,\mathbf{X})}[\ell(W, d(\mathbf{X}))] \\ &= \sum_{W \in \Omega_W} \int_{\mathbf{X} \in \Omega_x} \ell(W, d(\mathbf{X})) p(W, \mathbf{X}) \mathrm{d}\mathbf{X} \qquad (3.4) \\ &= \int_{\mathbf{X} \in \Omega_x} p(\mathbf{X}) [\sum_{W \in \Omega_W} \ell(W, d(\mathbf{X})) P(W|\mathbf{X})] \mathrm{d}\mathbf{X} \qquad (3.5) \\ &= \sum_{W \in \Omega_W} P(W) \int_{\mathbf{X} \in \Omega_x} \ell(W, d(\mathbf{X})) p(\mathbf{X}|W) \mathrm{d}\mathbf{X} \ , \qquad (3.6) \end{aligned}$$

where $\mathbf{E}_{(W,\mathbf{X})}[\,\cdot\,]$ denotes mathematical expectation with respect to the distribution of $(W, \mathbf{X})$. The above total risk can be used as a measure of the quality of decision rules. Usually the less the total risk, the better is the decision rule. In this framework, the issue of constructing an optimal decision rule becomes the following risk minimization problem:

$$\min_{d(\cdot) \in \mathcal{D}} r(d(\cdot)) = \min_{d(\cdot) \in \mathcal{D}} \int_{\mathbf{X} \in \Omega_x} p(\mathbf{X}) [\sum_{W \in \Omega_W} \ell(W, d(\mathbf{X})) P(W|\mathbf{X})] \mathrm{d}\mathbf{X}. \tag{3.7}$$

This optimization can be solved by minimizing the expression in the square brackets in the above equation. It is clear that the solution leads to the following optimal decision rule:

$$d_o(\mathbf{X}) = \arg \min_{d(\mathbf{X}) \in \Omega_W} \sum_{W \in \Omega_W} \ell(W, d(\mathbf{X})) P(W|\mathbf{X}), \tag{3.8}$$

which is also known as the *Bayes' decision rule*. The resulting minimum total risk,

$$r(d_o(\cdot)) = \int_{\mathbf{X}\in\Omega_x} p(\mathbf{X})[\sum_{W\in\Omega_W} \ell(W, d_o(\mathbf{X}))P(W|\mathbf{X})]\mathrm{d}\mathbf{X}, \tag{3.9}$$

is called the *Bayes' risk*. This risk value is the best that can be achieved if the distribution $p(W, \mathbf{X})$ is known.
In speech recognition, a reasonable option is to assume that every misclassification of $\mathbf{X}$ is equally serious, thereby resulting in the so-called *0-1 loss function*,

$$\ell(W, d(\mathbf{X})) = \begin{cases} 0 \text{ if } W = d(\mathbf{X}) \text{ (correct decision)} \\ 1 \text{ if } W \neq d(\mathbf{X}) \text{ (wrong decision)} \end{cases} \tag{3.10}$$

for $W \in \Omega_W$, $d(\mathbf{X}) \in \Omega_W$. Substituting (3.10) into (3.6), we obtain

$$r_{01}(d(\cdot)) = \sum_{W\in\Omega_W} P(W) \int_{\mathbf{X}\notin\Omega_x(W)} p(\mathbf{X}|W)\mathrm{d}\mathbf{X} \tag{3.11}$$

$$= 1 - \sum_{W\in\Omega_W} \int_{\mathbf{X}\in\Omega_x(W)} P(W)p(\mathbf{X}|W)\mathrm{d}\mathbf{X}. \tag{3.12}$$

Therefore, in the case of the 0-1 loss function, the total risk is the unconditional error probability which is apparently a good measure of the quality of decision rules for the ASR task. The optimal decision rule, $d_{01}(\cdot)$, under the *minimum classification error* criterion with the 0-1 loss function is then solved as $d_{\mathrm{MAP}}(\mathbf{X}) = \hat{W}$ such that

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W|\mathbf{X}) = \underset{W}{\operatorname{argmax}}\, p(\mathbf{X}|W) \cdot P(W) \tag{3.13}$$

which is also known as the *MAP decision rule*.
In summary, in constructing these optimal decision rules, it was assumed that complete prior information about the classes is known, i.e.:

1) the observation space Ω_x is given;

2) the loss function $\ell(W, d(\mathbf{X}))$ is given; and

3) the true PDF $p(W, \mathbf{X})$ or $p(\mathbf{X}|W)$ and $P(W)$ are known.

Under these assumptions, the optimality criterion is the minimization of the risk functional $r(d(\cdot))$, and the optimal decision rule is the Bayes' decision rule.

3.3 Adaptive Decision Rules Constructed from Training Samples

In practice, we know neither the *true* parametric form of the joint distribution $p(W, \mathbf{X})$ nor its *true* parameters. We shall say that we have *prior uncertainty* [79] in this case.

If we have some labeled *independent* training sample set, $\mathcal{X} = \{(W^i, \mathbf{X}^i); i = 1, 2, \cdots, n\}$, obtained by a series of *independent* experiments such that $(W^i, \mathbf{X}^i) \sim p(W, \mathbf{X})$ for the ASR task at hand or in mind, we can reduce the prior uncertainty by constructing a decision rule from $\mathcal{X}$. The decision rule $d(\cdot) = d(\mathbf{X}; \mathcal{X})$ based on the training set $\mathcal{X}$ and used to classify a random observation $\mathbf{X}$ that is *independent* of $\mathcal{X}$, is called an *adaptive decision rule* [79]. There are several principles that can be used for the construction of such rules.Two of them are briefly discussed in the following.

3.3.1 Plug-in Bayes' Decision Rules with Maximum-likelihood Density Estimate

3.3.1.1 What are Plug-in Bayes' Decision Rules?

The most popular family of adaptive decision rules might be the so-called *plug-in decision rules*. For this approach, let $\{\hat{P}(W), \hat{p}(\mathbf{X}|W)\}$ be any statistical estimators of true distributions $\{P(W), p(\mathbf{X}|W)\}$ based on the training sample $\mathcal{X}$. The *plug-in decision rule* [32] is the adaptive decision rule $d = \hat{d}_o(\mathbf{X})$ derived from the Bayesian decision rule (3.8) by substitution of the estimators $\{\hat{P}(W), \hat{p}(\mathbf{X}|W)\}$ for unknown true distributions $\{P(W), p(\mathbf{X}|W)\}$:

$$\hat{d}_o(\mathbf{X}) = \arg \min_{d(\mathbf{X}) \in \Omega_W} \sum_{W \in \Omega_W} \ell(W, d(\mathbf{X})) \hat{P}(W|\mathbf{X}), \tag{3.14}$$

where

$$\hat{P}(W|\mathbf{X}) = \frac{\hat{p}(\mathbf{X}|W)\hat{P}(W)}{\sum_W \hat{p}(\mathbf{X}|W)\hat{P}(W)}. \tag{3.15}$$

By varying the loss function and by using the different kinds of estimators, a fairly rich family of plug-in decision rules can be obtained. For example, adopting the 0-1 loss function will lead to the following plug-in decision rule, $\hat{d}_{MAP}(\mathbf{X}) = \hat{W}$, such that

$$\hat{W} = \operatorname*{argmax}_W \hat{P}(W|\mathbf{X}) = \operatorname*{argmax}_W \hat{p}(\mathbf{X}|W) \cdot \hat{P}(W) \tag{3.16}$$

which is also known as the *plug-in MAP decision rule*.
It can be shown [32] that the plug-in decision rule $\hat{d}_o(\cdot)$ in Eq. (3.14) minimizes the *plug-in risk* $\hat{r}(d(\cdot))$,

$$\hat{r}(d(\cdot)) = \sum_{W \in \Omega_W} \hat{P}(W) \int_{\mathbf{X} \in \Omega_x} \ell(W, d(\mathbf{X})) \hat{p}(\mathbf{X}|W) \mathrm{d}\mathbf{X}, \tag{3.17}$$

which is an estimate of the total risk using the *density plug-in estimator* $\{\hat{P}(W), \hat{p}(\mathbf{X}|W)\}$, i.e.,

$$\hat{d}_o(\cdot) = \arg \min_{d(\cdot) \in \mathcal{D}} \hat{r}(d(\cdot)) \ . \tag{3.18}$$

The minimum plug-in risk is then $\hat{r}(\hat{d}_o(\cdot))$.

3.3.1.2 Why Could Plug-in Bayes' Decision Rules Work?

As noted in [32], the plug-in risk $\hat{r}(\hat{d}_o(\cdot))$ of the plug-in Bayes' decision rule in Eq. (3.14), is often less than its total risk $r(\hat{d}_o(\cdot))$ and is even optimistically biased as an estimator of the Bayes' risk $r(d_o(\cdot))$:

Property: If the estimators $\{\hat{P}(W), \hat{p}(\mathbf{X}|W)\}$ are pointwise unbiased, then

$$\mathbf{E}[\hat{r}(\hat{d}_o(\cdot))] \leq r(d_o(\cdot)) \leq r(\hat{d}_o(\cdot)). \tag{3.19}$$

However, the usefulness of the plug-in Bayes' decision rule in Eq. (3.14) can be justified by the following theorem of *Bayes' risk consistency* [32]:

Theorem: (Bayes' risk consistency): If the estimators $\{\hat{P}(W), \hat{p}(\mathbf{X}|W)\}$ are strongly consistent, i.e., converge to the true distributions almost surely as the training sample size n increases ($n \rightarrow \infty$):

$$\hat{P}(W) \xrightarrow{a.s.} P(W), \;\; \hat{p}(\mathbf{X}|W) \xrightarrow{a.s.} p(\mathbf{X}|W), \;\; \text{for } W \in \Omega_W \text{ and } \mathbf{X} \in \Omega_x; \tag{3.20}$$

then the plug-in risk for the plug-in decision rule in Eq. (3.14) is a strongly consistent estimator of the Bayes' risk, i.e.,

$$\hat{r}(\hat{d}_o(\cdot)) \xrightarrow{a.s.} r(d_o(\cdot)) \; . \tag{3.21}$$

3.3.1.3 Implications on Parametric Models and Parameter Estimation

In practice, because of the constraints of the limited computational resources and training data, we always have to *assume* some parametric form for $p(W, \mathbf{X})$, e.g., via $p_\Lambda(\mathbf{X}|W)$ and $P_\Gamma(W)$. The parameter set (Λ, Γ) has to be *estimated* from the given training set $\mathcal{X}$ by using certain parameter estimation techniques. The above Bayes' risk consistency theorem tells us that it is often possible to construct plug-in procedures that are *Bayes' risk consistent* in the sense that the sequence of plug-in risks converges to the Bayes' risk as the training sets increase in size. However, there is an important assumption behind this argument, that is, the assumed distributions $p_\Lambda(\mathbf{X}|W)$ and $P_\Gamma(W)$ obey the parametric structure in question. In order to achieve a good approximation to reality, some flexible parametric models should be adopted. Currently, the most widely adopted and the most successful modeling approach to ASR is to use a set of hidden Markov models (HMMs) as the acoustic models of subword or whole-word units, and to use the statistical N-gram model or its variants as language models for words and/or word classes. The readers are referred to good tutorials in [87, 104, 65] and [58] for an introduction to the above approaches and their applications. By using the abovementioned plug-in MAP decision rule, it has been repetitively shown by experiments in the past three decades that given a large amount of *representative* training speech and text data, good statistical models of speech and language can be constructed to achieve a high performance for a wide range of ASR tasks. This has given the speech research community a certain level of confidence in

believing that the *Discrete HMM* (DHMM, [90]) and the Mixture Gaussian *Continuous Density HMM* (CDHMM, [63, 103, 64]), together with N-gram models [58], provide a good approximate parametric form for $p_\Lambda(\mathbf{X}|W)$ and $P_\Gamma(W)$, respectively. Although these models are apparently imperfect [102, 17, 36, 108], they are mathematically well-defined and capable of simultaneously modeling both the spectral and temporal variation in speech. They are also well thought of because they both fit into the framework of *finite state* representations [9, 88] of *knowledge sources* so that the speech recognition problem can be solved as a *network search* problem over a complex network representation of speech and language [101]. Based on the belief that these acoustic and language models are good approximates, the *maximum likelihood* (ML) estimate for the HMM parameters [11, 79, 63, 64, 39, 54] and N-gram model parameters [37, 61, 58] has been the most popular parameter estimation method. The widespread use of the plug-in MAP decision rule with the ML estimator can be justified by using the above Bayes' risk consistency theorem due to the following facts:

1) The ML estimators of $\{\Lambda, \Gamma\}$ are strongly consistent, unbiased, and efficient.

2) This can then be translated into the distribution consistency if the parametric forms of the $p_\Lambda(\mathbf{X}|W)$ and $P_\Gamma(W)$ are indeed correct.

According to our knowledge, it was Nadas [97] who first provided such an insight for the speech recognition community.

Of course, one can always argue that although the ML estimators $\hat{\Lambda}$ and $\hat{\Gamma}$ may be excellent estimators of Λ and Γ, there is no guarantee that $P_{\hat{\Gamma}}(W)$ and $p_{\hat{\Lambda}}(\mathbf{X}|W)$ are good guesses for $P(W)$ and $p(\mathbf{X}|W)$ because of the incorrect model assumptions. Nor is $\hat{d}_o(\cdot)$ necessarily a good approximation to $d_o(\cdot)$. The performance of the plug-in rules and other procedures should really be judged by the criterion of total risk, or by other criteria tied more directly to the classification accuracy than to the behavior of $(\hat{\Lambda}, \hat{\Gamma})$ as a *point estimator* for (Λ, Γ). This has motivated many studies in the past two decades aiming at a good alternative to ML training. One method is *minimum discrimination information* (MDI) training [21] which adjusts the HMM parameters to minimize the *discrimination information*, or *directed divergence*, between the assumed HMM distribution and the best possible distribution derived from the training data under certain constraints embedded in the training data. Unfortunately, no significant experimental results have been reported to show how MDI works in a speech recognition task. Another class of approaches is the so-called *discriminative training* method. Some of them, such as *maximum mutual information* (MMI) training [7], *conditional maximum likelihood estimate* (CMLE) [99], and *H-criteria* [38], aim indirectly at reducing the error rate of the speech recognizer on the training set. Other methods such as *corrective training* [8] and *minimum empirical classification error* training [22, 92, 67, 69] try to reduce the recognition error rate on training sample set in a more direct way. Among these approaches, the minimum empirical classification error (known as MCE) formulation proposed in [67] is, in my opinion, more theoretically sound, thus will be discussed briefly in the following.

3.3.2 Maximum-Discriminant Decision Rules Minimizing the Empirical Classification Error

3.3.2.1 What are Maximum-Discriminant Decision Rules?

Suppose one can define a *discriminant function* $g_\Lambda(\mathbf{X}; W)$ for each class W that characterizes the similarity between an observation $\mathbf{X}$ and the class W, where Λ is the set of classifier parameters to be estimated from the training data set $\mathcal{X}$. Naturally, the following *maximum-discriminant decision rule* $d(\cdot)$,

$$\hat{W} = \arg\max_{W} g_\Lambda(\mathbf{X}; W) \tag{3.22}$$

can be used to classify an unknown observation $\mathbf{X}$ into one of the M classes in Ω_W. The obvious criterion for estimating the classifier parameters Λ is to minimize the empirical classification error on the training sample set $\mathcal{X}$ defined as follows:

$$\bar{r}(d(\cdot)) = 1 - \frac{\text{number of correct classifications by } d(\cdot)}{\text{total number of sample observations on } \mathcal{X}} \ . \tag{3.23}$$

Now let $\mathcal{D}$ denote an arbitrary, but completely specified, collection of discriminant-based decision rules. A sample-based discriminant decision rule $\bar{d}(\cdot) \in \mathcal{D}$ will be called a *minimum misclassification* or *best-count* discriminant decision rule if it minimizes the sample error rate $\bar{r}(d(\cdot))$ among all discriminant decision rules $d(\cdot) \in \mathcal{D}$; that is, a best-count discriminant decision rule $\bar{d}(\cdot) \in \mathcal{D}$ satisfies

$$\bar{r}(\bar{d}(\cdot)) = \min_{d(\cdot)\in\mathcal{D}} \bar{r}(d(\cdot)) \ .$$

Similar to the case of the *density estimator*, it can be shown [33] that

$$\mathbf{E}[\bar{r}(\bar{d}(\cdot))] \leq \min_{d(\cdot)\in\mathcal{D}} r_{01}(d(\cdot)) \leq r_{01}(\bar{d}(\cdot)) \ .$$

So, $\bar{r}(\bar{d}(\cdot))$ is an optimistically biased estimator of the actual error rate of $\bar{d}(\cdot)$ and the least possible error rate.

3.3.2.2 Why Could Discriminant Approach Work?

The usefulness of the best-count discriminant approach can be justified by the following theorem, similar to the one in section 3.3.1.2, and also proved by Glick [33]:

Theorem: (Uniform Convergence) A discriminant decision rule will be called *m-convex* if its partition regions are sets in the finite field generated by some m measurable convex sets. As the sample size $n \to \infty$, the estimator $\bar{r}(d(\cdot))$ converges to $r_{01}(d(\cdot))$ *uniformly* over all discriminant decision rules $d(\cdot)$ in any collection $\mathcal{D}^*$ of m-convex discriminant decision rules; that is, the convergence is almost surely (a.s.),

$$\sup_{d(\cdot)\in\mathcal{D}^*} |\bar{r}(d(\cdot)) - r_{01}(d(\cdot))| \xrightarrow{a.s.} 0 \ .$$

This uniform convergence implies that the *best-count* discriminant $\bar{d}(\cdot) \in \mathcal{D}^*$ is asymptotically optimal in the sense of

$$r_{01}(\bar{d}(\cdot)) \rightarrow \min_{d(\cdot)\in\mathcal{D}^*} r_{01}(d(\cdot)) \quad \text{with probability one; and}$$
$$\bar{r}(\bar{d}(\cdot)) \rightarrow \min_{d(\cdot)\in\mathcal{D}^*} r_{01}(d(\cdot)) \quad \text{with probability one.}$$

If collection $\mathcal{D}^*$ contains any optimal discriminant decision rule $d^*(\cdot)$,

$$d^*(\cdot) = \arg \min_{any\ d(\cdot)} r_{01}(d(\cdot)) \ ,$$

then $\bar{d}(\cdot)$ is asymptotically optimal in the unrestricted sense, viz., strongly consistent in Bayes' risk. As pointed out in [33], this result is narrower than its parallel result for density estimates stated in the theorem in Section 3.3.1.2. It will be interesting to investigate how far the above result can be generalized to a wider range of discriminant functions.

3.3.2.3 Implications on the Choice of Discriminant Functions and the Practical Training Algorithms

The above theoretical result gives one confidence that if a proper form for the discriminant functions can be specified for the given pattern recognition problem, it is often possible to construct maximum-discriminant decision rules by estimating the classifier parameters under the criterion of minimum empirical classification error. Such decision rules are Bayes' risk consistent in the sense that the sequence of empirical risks converges to the Bayes' risk as the training sets increase in size. Of course, how to define an optimal form for the discriminant functions is application dependent and remains largely an open research problem. On the other hand, the good news is that the smooth MCE objective function proposed in [67] can approximate the empirical error rate for the design sample set arbitrarily closely. It can thus be used as the design criterion to be optimized by any gradient-based optimization methods. In the past decade, this MCE formulation has been extensively studied, refined, and successfully applied to solving many pattern recognition applications, see for example [72, 15] and the references therein.

3.3.3 Discussion

So far we have considered the following two strategies that have been used to construct a modern ASR system:

1) Using *plug-in MAP* as a decision rule for recognition decision, and ML as a criterion for the estimation of decision parameters.

2) Using *maximum discriminant* as a decision rule for recognition decision and *minimum empirical classification error* (MCE) as a criterion for the estimation of decision parameters.

The following conclusions may be drawn concerning these two strategies:

1) The asymptotic behavior of the first approach will depend on the appropriateness (in the sense of estimator consistency) of the parametric forms of the assumed distributions.

2) The asymptotic behavior of the second approach will depend on the choice of the discriminant function.

Theoretically speaking, it is not so clear yet which strategy is better for a moderately sized training set. However, in the past decade, it has been demonstrated by many research groups that when sufficient amount of *representative* training data are available, an ASR system constructed under the second principle can outperform its counterpart constructed under the first principle for many ASR applications.

3.4 Violations of Modeling Assumptions in ASR

3.4.1 Types of Distortions

The principles of the construction of the abovementioned optimal decision rule and adaptive decision rules are based on some assumptions which may be violated in practice. From the computational modeling point of view, there are three main distortion types that produce violations of assumptions summarized as follows [79]:

1) distortions caused by small-sample effects;

2) distortions of models or discriminant functions for training samples; and

3) distortions of trained models or discriminant functions for observations to be classified.

The distortions caused by small-sample effects are typical for all statistical plug-in procedures. They arise from the noncoincidence of the statistical estimates $\{\hat{P}(W), \hat{p}(\mathbf{X}|W)\}$ of probability characteristics and their true values $\{P(W), p(\mathbf{X}|W)\}$. We want to emphasize again that the plug-in decision rules described in previous section are asymptotically optimal only when

1) the training samples $\mathcal{X} = \{(W^i, \mathbf{X}^i); i = 1, 2, \cdots, n\}$ are collected by a series of *independent* experiments such that $(W^i, \mathbf{X}^i) \sim p(W, \mathbf{X})$, or more intuitively speaking, $\mathcal{X}$ should be *representative* enough with respect to the true distribution of the testing data $\mathbf{X}$; and

2) training sample size $n \to \infty$, i.e., there is sufficient amount of training data available.

In practice, the training sample set $\mathcal{X}$ always has a finite size (i.e., $n < \infty$), and in many cases, is possibly also not representative enough. The random deviations of statistical estimates $\{\hat{P}(W) - P(W),\ \hat{p}(\mathbf{X}|W) - p(\mathbf{X}|W)\}$ can then produce significant increases of risk. As for the small-sample effects for discriminant-based approach, it is intuitively obvious that a small training error on a small set of (possibly not so representative) training samples does not necessarily guarantee a small test error. So, the design and/or collection of the training samples become very critical. The key is to make the samples in $\mathcal{X}$ follow the intended distribution $p(W, \mathbf{X})$ as closely as possible. Otherwise, some more intelligent ways of using the available training data must be developed.

As for the distortions of the models or discriminant functions for the training samples, they can be caused by the wrong assumptions and/or inflexible parametric forms of the model or discriminant function; the misclassification of training samples; outliers in training samples, etc. They will cause both *modeling error* and *estimation error*. To cope with these problems, better models or discriminant functions need to be found and techniques need to be designed for robust learning from data.

The biggest problem for ASR might be caused by the third type of distortion. In most real applications, there always exists some form of mismatch which causes a distortion between the trained models or discriminant functions and the test data. These mismatches, some of them identified in Figure 3.1, may arise from inter- and intra-speaker variabilities; transducer, channel and other environmental variabilities; and many other phonetic and linguistic effects caused by mismatch in training and testing task definitions. How to achieve the performance robustness in this context has become one of the most active research areas in ASR in the past decade.

3.4.2 Towards Adaptive and Robust ASR

From the above analysis and discussion, it is quite clear that in order to design an automatic speech recognizer that works well for different tasks and speakers over unexpected and possibly adverse conditions, all of the above three distortion types need to be appropriately treated. One of the effective ways to improve ASR robustness is to find invariant (or robust) features so as to minimize the observation variability caused by the different types of interfering factors and the possible mismatch between training and testing conditions. Even though some features have been shown less affected by a certain type of distortion, such as linear microphone or channel effect, no feature has yet been discovered that is invariant across all adverse acoustic conditions. Further research in front-end signal processing and feature extraction is definitely needed to improve on the currently "standard" acoustic analysis for ASR [43]. Once the feature extraction method is fixed, another traditional approach to robust speech recognition is to develop better modeling and learning techniques that have a good generalization capability. In addition, four major classes of statistical techniques to improve ASR robustness can be defined:

1) adapting recognizer parameters to new operating conditions using adaptation and/or testing data;

2) modifying signal, feature, or recognizer parameters using only the utterance to be recognized to reduce the mismatch between the training and testing conditions;

3) using robust decision strategies; and

4) possible combinations of the above techniques.

Along these lines, many techniques have been developed and are reviewed from different perspectives in, for example [66, 23, 1, 35, 70, 25, 112, 26, 85, 86, 51]. Readers are referred to these reviews for a rich picture of the field and the references therein for the details of the different techniques. In the remaining part of the chapter, I will briefly review two technologies, namely, recognizer parameter adaptation and robust decision rules, that were developed in the past decade to cope with the above problems. The selection of materials is guided by the consideration that discussions can be made in a relatively more rigorous way from the viewpoint of previously discussed decision theoretic formulations for the ASR problem.

3.5 Improving Adaptive Decision Rules via Decision Parameter Adaptation

3.5.1 Decision Parameter Adaptation for Stationary Operating Conditions

If the operating condition of a speech recognizer is stationary, then there must exist a true distribution $p(W, \mathbf{X})$. Suppose the training data $\mathcal{X}$ is not representative enough so that the recognizer constructed using the design principles discussed previously does not work so well for the testing data from $p(W, \mathbf{X})$. If the application scenario allows, a straightforward solution to improving the adaptive decision rules is to collect additional training data $\mathcal{X}_a = \{(W_a^i, \mathbf{X}_a^i); i = 1, 2, \cdots, N_a\}$ (known as adaptation data) in a specific testing condition such that $(W_a^i, \mathbf{X}_a^i) \sim p(W, \mathbf{X})$, and then to adapt the recognizer parameters accordingly to work better in the prescribed scenario. Depending on which design principle was used to construct the speech recognizer from the training sample $\mathcal{X}$, there are naturally two *goals of adaptation*, namely ML and MCE, for adapting recognizer parameters using $\mathcal{X}_a$. By doing so, the previous discussions about the asymptotic properties of the two design principles remain true, thus the performance of the adapted recognizer can approach the matched-condition performance with the increasing amount of adaptation data $\mathcal{X}_a$. However, in order to hold and/or improve ASR performance with a small amount of adaptation data, special measures must be taken to deal with the problem of estimating a large number of parameters from sparse data.

3.5.1.1 Adaptation for Plug-in Decision Rules

Consistent with the first design principle discussed in Section 3.3, many successful adaptation techniques have been developed in the past decade to cope with the possible problem of mismatches between training and testing conditions. Because we have already given an overview of these techniques in [86] recently, I just add two more remarks here to supplement the detailed discussions in [86]:

Remark 1: To deal with the sparse data problem, two strategies have been successfully used. One is the approach of *regularization* and another is the approach of *imposing constraints* to reduce the degrees of freedom for parameter estimation. The popular Bayesian point estimate such as the MAP estimate is an example of the former, while the transformation-based approach is an example of the latter. So, the MAP estimate is sometimes also referred to as *maximum penalized likelihood* estimate. Of course, the above two strategies can be simultaneously used to deal with the sparse data problem.

Remark 2: Unsupervised adaptation remains largely an unsolved research problem. Transformation-based unsupervised adaptation works sometimes just because the transformations are possibly shared by different speech units, thus the consequence of the wrong supervision is not as severe as in other approaches without using the mechanism of parameter tying or sharing.

3.5.1.2 Adaptation for Maximum-Discriminant Decision Rules

In contrast with the extensive researches under the first design principle, less efforts have been devoted to develop techniques for decision parameter adaptation which is consistent with the second design principle discussed in Section 3.3. A study on MCE adaptation of CDHMM parameters was first carried out by authors of [89]. Several follow-up studies were also reported by other research groups [93, 115, 82]. A more recent one was reported in [83] and demonstrated that direct MCE adaptation for MCE-trained HMM parameters works well when sufficient (*w.r.t.* the number of parameters being adapted) amount of adaptation data are available. However, when only small amount of adaptation data are available, direct MCE adaptation of HMM parameters does not work so well. The lack of an efficient adaptation algorithm for MCE-trained seed models might be one of the main reasons why the MCE training has not been widely used yet to construct an ASR system for applications in which decision parameter adaptation is required.

In the past several years, there have been some efforts to develop discriminative linear regression adaptation techniques under different criteria and notions such as MCE [12], maximum scaled likelihood [120], maximal rank likelihood [29], MMI (maximum mutual information) [118], and CML (conditional maximum likelihood) [40]. Interestingly, although all of them are developed with the aim of an efficient discriminative adaptation, they have only been applied to adapting the ML-trained seed models. No results have been reported yet how they work for the adaptation of the discriminatively trained seed models. In my opinion, this is a more desirable scenario to apply discriminative adaptation, because the consistent criteria are used

in both seed model training and the succeeding adaptation, and hopefully a better performance can be achieved in this way. It is this fact that motivates us to perform a study as reported in [123]. In [123], we have presented a formulation of minimum classification error linear regression (MCELR) for adaptation of Gaussian mixture CDHMM parameters. We demonstrate that the MCELR can be used to adapt the MCE-trained HMM parameters under a consistent criterion. In a supervised speaker adaptation application, we observe that such adapted models perform better than the ones adapted using maximum likelihood linear regression (MLLR) from the ML-trained seed models. Further studies are needed to explore MCELR's behavior for long-term adaptation using increasing amount of adaptation data.

In addition to using the above minimum *empirical* classification error criterion for decision parameters adaptation, one can also adopt another criterion called *minimum expected classification error* as defined in the following:

$$R(\Lambda) = \sum_{W \in \Omega_W} \int_{\mathbf{X} \in \Omega_x} \ell_\Lambda(W, d(\mathbf{X})) p(W, \mathbf{X}) \mathrm{d}\mathbf{X} \ , \tag{3.24}$$

where $\ell_\Lambda(W, d(\mathbf{X}))$ is a loss function characterized by the decision rule parameters Λ. Apparently, the above objective function $R(\Lambda)$ is an underdefined functional because the true distribution $p(W, \mathbf{X})$ is unknown. However, by using the *stochastic approximation* method suggested in the 1950s by Robbins and Monroe [107], the functional $R(\Lambda)$ can be minimized with respect to the parameters Λ by using the testing data $\{(W^t, \mathbf{X}^t); t = 1, 2, \cdots\}$ drawn from $p(W, \mathbf{X})$ as follows:

$$\Lambda_{t+1} = \Lambda_t - \alpha_t \nabla \ell_\Lambda(W^t, d(\mathbf{X}^t)) \ . \tag{3.25}$$

It can be proven that this method is consistent under very general conditions on the gradient $\nabla \ell_\Lambda(W^t, d(\mathbf{X}^t))$ and the schedule of the learning rate α_t. Historically, this approach was independently proposed and developed for pattern recognition application by Amari [5] and Tsypkin [116, 117] respectively. Again, a smooth loss function $\ell_\Lambda(W, d(\mathbf{X}))$ was proposed in [67] to make the above procedure practically useful. Apparently, the above general learning principle can be used for supervised on-line adaptation of the decision parameters Λ. However, it converges in probability which means that the algorithm converges only after a large amount of samples are used. This makes the approach more suitable for long-term adaptation.

3.5.2 Decision Parameter Adaptation for Slowly Changing Operating Conditions

Most of the existing adaptation algorithms treat the individual data block $(W_a^i, \mathbf{X}_a^i)$ of the available adaptation data $\mathcal{X}_a$ as equally important, thus are valid only in a stationary operating condition for estimating stationary parameters. However, in many real speech recognition applications, the statistical characteristics of the observation data undergo gradual changes due to many possible factors such as the changing speaking behavior of a speaker, the changing operating environment, the changing transmission channel, etc. The problem of parametric learning with such slowly

changing operating conditions is to estimate time-varying decision rule parameters. In such cases, different data segments often correspond to different parameter values. In order to continuously track the variations of the model parameters corresponding to the new data, some *forgetting mechanisms* are needed to reduce the effect of past observations relative to the new input data. This makes the on-line learning algorithm with forgetting capabilities a natural choice for making the recognition system capable of continuously adjusting to a new operating condition without the requirement of storing a large set of previously used training data. The series of Bayesian learning algorithms for CDHMM parameters developed in [48, 49, 53] are designed for dealing with the slow change of the operating conditions from utterance to utterance, while the algorithms developed in [80, 81, 16, 78, 2, 124] can be operated in a frame-synchronous fashion so that they are presumably able to deal with the within-utterance nonstationarity. If the forgetting mechanism is disabled, the above algorithms can also be used to adapt decision rule parameters for a stationary operating condition. As a final remark, the above algorithms are developed to adapt the parameters of the plug-in MAP decision rule. How to adapt the parameters of the maximum-discriminant decision rule in a nonstationary operating condition remains an interesting open problem.

3.5.3 Decision Parameter Adaptation for Switching Operating Conditions

When an ASR system has to be operated under rapidly switching conditions, the above adaptation algorithms can not be applied. If the nonstationary operating condition can be approximated by a finite number of different stationary conditions, then a simple solution could be imagined. An offline condition-clustering can be performed first and an individual recognizer is then constructed for each cluster. Given an unknown utterance to be recognized, the most similar condition-cluster can be identified and the associated recognizer can be used to recognize the unknown utterance. The traditional technique of speaker adaptation via speaker clustering and selection is a good example of this strategy [90, 74]. However, if the current operating condition is not similar to any single training condition yet shares certain characteristics with some training conditions, then a strategy of *adaptive model fusion* can be adopted as for example in [28, 122, 45, 27, 42, 75, 52]. All of these works share the similarity in the general sense that they

- first prepare offline a set of models from training data, and then
- fuse adaptively, by using the information embedded in the utterance to be recognized, a set of new models which hopefully is more "appropriate" to the testing utterance, and finally
- re-recognize the testing utterance again.

Although the above approaches have mainly been developed and studied for dealing with the speaker variability, the same idea of offline variability decomposition and

online adaptive model fusion can be further explored to deal with other aspects of robust ASR.

3.5.4 Discussion

So far, I have briefly discussed several strategies for decision parameter adaptation in three types of operating conditions. It is clear that the greatest challenge comes from those applications which only involve a couple of utterances, but every utterance involves a distinct yet complicated "distortion channel" from the intended message a speaker want to convey to the received signal of a speech recognizer. After all of the above adaptation techniques have been considered, another strategy namely *robust decision rule* can always be tried out to see whether performance robustness can be further improved. In the remaining part of the chapter, This strategy is explained in detail.

3.6 Robust Decision Rules

3.6.1 Decision Rule Robustness

Intuitively speaking, a decision strategy (rule) is called robust if it is not very sensitive to the previously discussed prior uncertainty (or distortions). More formally, let $d(\cdot) = d(\mathbf{X}; \mathcal{X})$ be an arbitrary decision rule constructed under some hypothetical model $\mathcal{M}_0$; where $d(\mathbf{X}) \in \Omega_W$ is the class to which the observation $\mathbf{X} \in \Omega_x$ will be assigned, and $\mathcal{X}$ is a training sample set used for the construction of the decision rule. Let $\mathcal{M}_\epsilon$ denote an arbitrary admissible distorted data model for the distortion types discussed in Section 3.4, where $\epsilon \geq 0$ is used to characterize the distortion level. Let $\mathcal{M}_\epsilon^*$ denote the set of admissible distorted data models. The classification performance of the decision rule $d(\cdot)$ in a situation where data are fitted to the distorted model $\mathcal{M}_\epsilon \in \mathcal{M}_\epsilon^*$ will be characterized by the risk functional:

$$r_\epsilon(d(\cdot)) = \mathbf{E}[\ell(W, d(\mathbf{X}))] \ ,$$

where $\mathbf{E}[\cdot]$ denotes the expectation with respect to the probability distribution of $(W, \mathbf{X})$ corresponding to the distorted model $\mathcal{M}_\epsilon \in \mathcal{M}_\epsilon^*$. Let us call the functional

$$r_+ = r_+(d(\cdot)) = \sup_{\mathcal{M}_\epsilon \in \mathcal{M}_\epsilon^*} r_\epsilon(d(\cdot))$$

the *guaranteed (upper) risk* [79] for the decision rule $d(\cdot)$ in the presence of distortions $\mathcal{M}_\epsilon \in \mathcal{M}_\epsilon^*$. If we know the distribution of $\mathcal{M}_\epsilon$ on $\mathcal{M}_\epsilon^*$, we can further define the following functional

$$\tilde{r} = \tilde{r}(d(\cdot)) = \mathbf{E}[r_\epsilon(d(\cdot))] \ ,$$

where $\mathbf{E}[\cdot]$ denotes the expectation with respect to the distribution of $\mathcal{M}_\epsilon$ on $\mathcal{M}_\epsilon^*$. We call $\tilde{r}(d(\cdot))$ the *overall risk*. Apparently, both $r_+(d(\cdot))$ and $\tilde{r}(d(\cdot))$ can be used as optimality criteria in searching for *robust (with respect to distortions $\mathcal{M}_\epsilon^*$) decision rules*. A decision rule $d^*(\mathbf{X};\mathcal{X})$ with the minimal value of the guaranteed risk for all admissible distortions:

$$d^*(\cdot) = \operatorname*{argmin}_{d(\cdot)} r_+(d(\cdot)) \tag{3.26}$$

is referred to as a *minimax decision rule*. A decision rule $\tilde{d}(\mathbf{X};\mathcal{X})$ with the minimal value of the overall risk for all admissible distortions:

$$\tilde{d}(\cdot) = \operatorname*{argmin}_{d(\cdot)} \tilde{r}(d(\cdot)) \tag{3.27}$$

is referred to as a *predictive decision rule*.
The construction of these robust decision rules will depend on how the admissible distortions $\mathcal{M}_\epsilon \in \mathcal{M}_\epsilon^*$ are defined, and also, for the case of the predictive decision rule, the distribution of the distortion on $\mathcal{M}_\epsilon^*$. In the following two subsections, I show two examples of such robust decision rules, namely *minimax decision rule* and *Bayesian predictive decision rule*, respectively. Both of them assume that

- the distributions $p(\mathbf{X}|W)$ and $P(W)$ are known up to some specifiable parameters in the forms of $p_\Lambda(\mathbf{X}|W)$ and $P_\Gamma(W)$;
- the true parameters of these distributions, Λ and Γ, lie in a neighborhood of the estimated (or hypothetical) ones; therefore,
- the prior uncertainty can be modeled by defining an *uncertainty neighborhood* of the model parameters Λ, Γ, and/or possibly a distribution of model parameters $p(\Lambda,\Gamma)$ on this uncertainty neighborhood.

With these assumptions, the specific minimax decision rule and predictive decision rule can be constructed accordingly to satisfy some desired robustness properties.

3.6.2 Minimax Classification Rule

Let $\eta_\epsilon(\Lambda_0,\Gamma_0)$ denote the *uncertainty neighborhood* of the true model parameters Λ,Γ, i.e., $(\Lambda,\Gamma) \in \eta_\epsilon(\Lambda_0,\Gamma_0)$, where Λ_0,Γ_0 are model parameters estimated from the training data $\mathcal{X}$, and ϵ can be viewed as a generic parameter to characterize the degree of the distortion. Then, we have

$$\mathcal{M}_\epsilon^* = \{p_\Lambda(\mathbf{X}|W), P_\Gamma(W) \mid (\Lambda,\Gamma) \in \eta_\epsilon(\Lambda_0,\Gamma_0)\}\,,$$

where $\mathcal{M}_\epsilon^*$ is the set of distorted models, and

$$r_+ = r_+(d(\cdot)) = \sup_{(\Lambda,\Gamma)\in\eta_\epsilon(\Lambda_0,\Gamma_0)} \sum_{W\in\Omega_W} P_\Gamma(W) \int_{\mathbf{X}\in\Omega_x} \ell(W,d(\mathbf{X}))p_\Lambda(\mathbf{X}|W)\mathrm{d}\mathbf{X}\,.$$

To construct a minimax decision rule which minimizes the above guaranteed risk $r_+(d(\cdot))$ is not a easy task. In practice, some more relaxed criteria have to be

adopted. One possibility is to use the upper bound of $r_+(d(\cdot))$, which we denote $r_{++}(d(\cdot))$,

$$r_{++} = r_{++}(d(\cdot)) = \sum_{W \in \Omega_W} \int_{\mathbf{X} \in \Omega_x} \sup_{(\Lambda,\Gamma) \in \eta_\epsilon(\Lambda_0,\Gamma_0)} \ell(W, d(\mathbf{X})) p_\Lambda(\mathbf{X}|W) P_\Gamma(W) \mathrm{d}\mathbf{X} \ .$$

To simplify our discussion, we assume that we do not consider the uncertainty of $P(W)$ thereafter and use $P_{\Gamma_0}(W)$ as the language model, with Γ_0 being the set of language model parameters estimated from the training text data. By using the (0,1)-loss function, we then have

$$r_{++} = r_{++}(d(\cdot)) = \sum_{W \in \Omega_W} P_{\Gamma_0}(W) \int_{\mathbf{X} \notin \Omega_x(W)} \sup_{\Lambda \in \eta_\epsilon(\Lambda_0)} p_\Lambda(\mathbf{X}|W) \mathrm{d}\mathbf{X} \ . \quad (3.28)$$

A decision rule which minimizes the above $r_{++}(d(\cdot))$ is as follows:

$$d_{++}(\mathbf{X}) = \arg\max_W [P_{\Gamma_0}(W) \max_{\Lambda \in \eta_\epsilon(\Lambda_0)} p_\Lambda(\mathbf{X}|W)] \ . \quad (3.29)$$

This is the so-called *minimax decision rule* which was first studied by Merhav and Lee in [95]. It can be solved in two steps. First, we estimate the underlying parameters using the ML approach within each neighborhood $\eta_\epsilon(\Lambda_0^{(W)})$, i.e.

$$\hat{\Lambda}_W = \arg \max_{\Lambda \in \eta_\epsilon(\Lambda_0^{(W)})} p_\Lambda(X|W), \quad (3.30)$$

where $\Lambda_0^{(W)}$ denotes pre-trained model parameters for word W. Then we apply the plug-in MAP decision rule with $\hat{\Lambda}_W$ replacing the original $\Lambda_0^{(W)}$. Therefore, conceptually, the minimax decision rule described in Eq. (3.29) can be viewed as a procedure which modifies the plug-in MAP decoder shown in Eq. (3.1) with an extra step as in Eq. (3.30) to find a modified point estimate in the neighborhood $\eta_\epsilon(\Lambda_0) = \{\eta_\epsilon(\Lambda_0^{(W)})\}$ of the original classifier parameters $\Lambda_0 = \{\Lambda_0^{(W)}\}$.
The above robust minimax classification rule makes no assumption about the form of the distortion. However, its efficacy does depend on an appropriate specification of the parameter uncertainty neighborhood $\eta_\epsilon(\Lambda_0) = \{\eta_\epsilon(\Lambda_0^{(W)})\}$. In the past several years, some other specific techniques have also been developed to implement the above minimax decision rule in HMM-based ASR systems [96, 51, 62, 3]. They are shown to be effective in dealing with noisy speech recognition and the mismatch caused by different recording conditions.
There are however other possibilities to model the admissible distortions $\mathcal{M}_\epsilon^*$. For example, if we use

$$\mathcal{M}_\epsilon^* = \{p_\Lambda(\mathbf{X}|W) \mid \Lambda = \mathcal{T}_\vartheta(\Lambda_0)\} \ ,$$

where $\mathcal{T}_\vartheta(\Lambda_0)$ denotes a specific transformation of Λ_0 with parameters ϑ. In this way, the uncertainty of Λ can be characterized by the uncertainty of ϑ. Then the *minimax decision rule* with respect to the above $\mathcal{M}_\epsilon^*$ will be

$$d_{++}(\mathbf{X}) = \arg\max_W [P_{\Gamma_0}(W) \max_\vartheta p(\mathbf{X}|W, \Lambda = \mathcal{T}_\vartheta(\Lambda_0))] \ .$$

The so-called *model-space stochastic matching* method described in [109, 113] can be theoretically justified in this way.

3.6.3 Bayesian Predictive Classification Rule

As I discussed before, minimax classification tries to handle the worst case mismatch by assuming a uniform distribution in the uncertainty neighborhood for all possible deviations from the nominal parameters, Λ_0. Instead of assigning another point estimate, $\hat{\Lambda}$, as done in the minimax classification rule discussed above, one can also *average out* the effect of the possible modeling and estimation errors by assuming a general prior PDF for Λ to characterize the parameter variability while making classification decisions. In this way, a new robust decision strategy can be derived and is often referred to as a *Bayesian predictive classification* (BPC) rule [98, 50, 51].

Let us consider the uncertainty of the model parameters (Λ, Γ) by treating them as if they were random. Our prior knowledge about (Λ, Γ) is assumed to be summarized in a known joint *a priori* density $p(\Lambda, \Gamma|\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)})$, with $\Lambda \in \Omega_\Lambda$ and $\Gamma \in \Omega_\Gamma$; where Ω_Λ and Ω_Γ denote the admissible regions of Λ and Γ, and $\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)}$ are the sets of parameters of the prior PDF (often referred to as *hyperparameters*) which are assigned values by the investigator. Such prior information may, for example, come from subject matter considerations and/or from previous experiences. For the simplicity of the following discussion, let us further assume that

$$p(\Lambda, \Gamma|\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)}) = p(\Lambda|\varphi_\Lambda^{(0)}) \cdot p(\Gamma|\varphi_\Gamma^{(0)}) \ .$$

Given a training set $\mathcal{X}$ as described at the beginning of Section 3.3, the uncertainty about Λ, Γ can be reduced by evolving $p(\Lambda, \Gamma|\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)})$. Apparently, there are many ways to evolve $p(\Lambda, \Gamma)$ which depend on the purpose of the modeling in mind, the knowledge/information sources used, and the possible constraints imposed [48, 49, 53]. It is at this point that our proposal departs from the conventional treatment in statistics. Conventionally, $p(\Lambda, \Gamma|\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)})$ is evolved by constructing the following posterior PDF:

$$\begin{aligned} p(\Lambda, \Gamma|\mathcal{X}) &= \frac{p(\mathcal{X}|\Lambda, \Gamma) \cdot p(\Lambda, \Gamma|\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)})}{\int_{\Omega_\Lambda} \int_{\Omega_\Gamma} p(\mathcal{X}|\Lambda, \Gamma) \cdot p(\Lambda, \Gamma|\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)}) \mathrm{d}\Lambda \mathrm{d}\Gamma} \\ &= p(\Lambda|\mathcal{X}) \cdot p(\Gamma|\mathcal{X}) \ , \end{aligned} \tag{3.31}$$

where

$$p(\Lambda|\mathcal{X}) = \frac{p(\mathcal{X}|\Lambda) \cdot p(\Lambda|\varphi_\Lambda^{(0)})}{\int_{\Omega_\Lambda} p(\mathcal{X}|\Lambda) \cdot p(\Lambda|\varphi_\Lambda^{(0)}) \mathrm{d}\Lambda} \ ,$$

$$p(\Gamma|\mathcal{X}) = \frac{p(\mathcal{X}|\Gamma) \cdot p(\Gamma|\varphi_\Gamma^{(0)})}{\int_{\Omega_\Gamma} p(\mathcal{X}|\Gamma) \cdot p(\Gamma|\varphi_\Gamma^{(0)}) \mathrm{d}\Gamma} \ .$$

This posterior PDF $p(\Lambda, \Gamma|\mathcal{X})$ includes all of the information inherited from the prior knowledge $p(\Lambda, \Gamma|\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)})$ and learned from the training data $\mathcal{X}$. Conventionally,

one derives a *point estimate* $\hat{\Lambda}, \hat{\Gamma}$ from $p(\Lambda, \Gamma|\mathcal{X})$ (e.g., MAP estimate) and then use the plug-in MAP decision rule for recognition. The conventional plug-in MAP decision rule based on the ML estimate of the model parameters Λ, Γ can be treated as a special case of the above MAP estimate with a non-informative prior.
However, in our proposal, we do not use $p(\Lambda, \Gamma|\mathcal{X})$ mechanically. Instead, we adopt a more flexible *empirical Bayes* approach in which a specific parametric PDF

$$p(\Lambda, \Gamma|\varphi_\Lambda, \varphi_\Gamma) = p(\Lambda|\varphi_\Lambda) \cdot p(\Gamma|\varphi_\Gamma)$$

is used to represent our uncertainty about Λ, Γ after observing $\mathcal{X}$. The intractability of directly calculating $p(\Lambda, \Gamma|\mathcal{X})$ for the popular models in ASR such as HMMs is not the only reason for the above proposal. A more important reason is that using $p(\Lambda, \Gamma|\varphi_\Lambda, \varphi_\Gamma)$, instead of $p(\Lambda, \Gamma|\mathcal{X})$, to represent the *prior uncertainty* (with respect to recognizing $\mathbf{X}$), provides a flexible way to incorporate and make use of other knowledge sources which may be available (in addition to $\mathcal{X}$ and $p(\Lambda, \Gamma|\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)})$) and/or even to consider the modeling intention. For example, the set of hyperparameters, $\varphi_\Lambda, \varphi_\Gamma$, could be estimated from training data $\mathcal{X}$, or specified using some empirical reasoning, or their combination [50, 51, 60, 61]. Furthermore, if the training data $\mathcal{X}$ is not *representative* enough, then such learned $p(\Lambda, \Gamma)$ might not be informative enough to help recognize $\mathbf{X}$. In this case, before invoking the recognition process, we can first improve $p(\Lambda, \Gamma)$ by using the information embedded in the observation $\mathbf{X}$ itself. In any case, we use $p(\Lambda, \Gamma|\varphi_\Lambda, \varphi_\Gamma)$ generically to represent our *prior uncertainty* about Λ, Γ just before making the decision on $\mathbf{X}$. In this way, we are essentially considering the following admissible distorted set of data model $\mathcal{M}_\epsilon^*$:

$$\mathcal{M}_\epsilon^* = \{p_\Lambda(\mathbf{X}|W), P_\Gamma(W) \mid (\Lambda, \Gamma) \sim p(\Lambda, \Gamma|\varphi_\Lambda, \varphi_\Gamma); \Lambda \in \Omega_\Lambda, \Gamma \in \Omega_\Gamma\} ,$$

where we can view the ϵ as a parameter to characterize the broadness of the distribution $p(\Lambda, \Gamma|\varphi_\Lambda, \varphi_\Gamma)$, or equivalently, the degree of the distortion. Based on the above $\mathcal{M}_\epsilon^*$, the *overall risk* is $\tilde{r}(d(\cdot))$,

$$\begin{aligned}
\tilde{r}(d(\cdot)) &= \mathbf{E}_{(W,\mathbf{X})}\mathbf{E}_{(\Lambda,\Gamma)}[\ell(W, d(\mathbf{X}))] \\
&= \sum_{W \in \Omega_W} \int_{\mathbf{X} \in \Omega_x} \int_{\Omega_\Lambda} \int_{\Omega_\Gamma} \ell(W, d(\mathbf{X})) p(W, \mathbf{X}|\Lambda, \Gamma) p(\Lambda, \Gamma|\varphi_\Lambda, \varphi_\Gamma) \mathrm{d}\Lambda \mathrm{d}\Gamma \mathrm{d}\mathbf{X} \\
&= \sum_{W \in \Omega_W} \int_{\mathbf{X} \in \Omega_x} \ell(W, d(\mathbf{X})) \tilde{p}(\mathbf{X}|W) \tilde{P}(W) \mathrm{d}\mathbf{X} ,
\end{aligned}$$

where

$$\tilde{p}(\mathbf{X}|W) = \int_{\Omega_\Lambda} p(\mathbf{X}|W, \Lambda) p(\Lambda|\varphi_\Lambda) \mathrm{d}\Lambda , \tag{3.32}$$

$$\tilde{P}(W) = \int_{\Omega_\Gamma} p(W|\Gamma) p(\Gamma|\varphi_\Gamma) \mathrm{d}\Gamma \tag{3.33}$$

are called *predictive densities* [4, 31, 106], because we can view $p(\Lambda|\varphi_\Lambda), p(\Gamma|\varphi_\Gamma)$ as a function of training samples $\mathcal{X}$. Then, under the (0,1)-loss function, the predictive

decision rule which minimizes the above $\tilde{r}(d(\cdot))$ is as follows:

$$\tilde{d}(\mathbf{X}) = \underset{W}{\operatorname{argmax}}\, \tilde{P}(W, \mathbf{X}) = \underset{W}{\operatorname{argmax}}\, \tilde{p}(\mathbf{X}|W) \cdot \tilde{P}(W) \ . \tag{3.34}$$

This decision rule $\tilde{d}(\cdot)$ will be referred to as the *Bayesian predictive classification (BPC) rule*. Three key issues thus arise in BPC, namely,

1) the definition of the prior density $p(\Lambda, \Gamma|\varphi_\Lambda, \varphi_\Gamma)$ for modeling the uncertainty of the model parameters Λ, Γ;

2) the specification of the hyperparameters, $\varphi_\Lambda, \varphi_\Gamma$; and

3) the evaluation of the predictive density.

In the past several years, some specific techniques have been developed to address the above issues, and some encouraging results have been obtained. Readers are referred to [50, 51, 60, 61]) for details.
In extending the above-formulated BPC approach, there are two opposite directions which can be pursued. One direction is to use a structure model for modeling parameter uncertainty. For example, we can use

$$\mathcal{M}_\epsilon^* = \{p_\Lambda(\mathbf{X}|W), P_\Gamma(W) \mid \Lambda = \mathcal{T}_{\vartheta_\Lambda}(\Lambda_0), \Gamma = \mathcal{T}_{\vartheta_\Gamma}(\Gamma_0), (\vartheta_\Lambda, \vartheta_\Gamma) \sim p(\vartheta_\Lambda, \vartheta_\Gamma)\} \ ,$$

where $\mathcal{T}_{\vartheta_\Lambda}(\Lambda_0)$ and $\mathcal{T}_{\vartheta_\Gamma}(\Gamma_0)$ denote specific transformations of Λ_0 and Γ_0 with parameters ϑ_Λ and ϑ_Γ respectively. In this way, the uncertainty of Λ, Γ can be characterized by the uncertainty of $\vartheta_\Lambda, \vartheta_\Gamma$. The BPC decision rule in (3.34) can then be modified using the following predictive PDFs:

$$\tilde{p}(\mathbf{X}|W) = \int_{\Omega_{\vartheta_\Lambda}} p(\mathbf{X}|W, \Lambda = \mathcal{T}_{\vartheta_\Lambda}(\Lambda_0)) p(\vartheta_\Lambda) \mathrm{d}\vartheta_\Lambda \ ,$$

$$\tilde{P}(W) = \int_{\Omega_{\vartheta_\Gamma}} p(W|\Gamma = \mathcal{T}_{\vartheta_\Gamma}(\Gamma_0)) p(\vartheta_\Gamma) \mathrm{d}\vartheta_\Gamma \ .$$

The above issue of prior specification will then be translated into the specification of $p(\vartheta_\Lambda)$ and $p(\vartheta_\Gamma)$. Readers are referred to [59] for a recent work along this line of thought. Another possible direction to pursue is to go beyond the *model parameter uncertainty* by considering the admissible distorted densities at the distribution level,

$$\begin{aligned} \mathcal{M}_\epsilon^* = \{&p(\mathbf{X}|W), P(W) \mid p(\mathbf{X}|W) = (1-\epsilon_1) p_{\Lambda_0}(\mathbf{X}|W) + \epsilon_1 h_1(\mathbf{X}|W), \\ &P(W) = (1-\epsilon_2) P_{\Gamma_0}(W) + \epsilon_2 h_2(W);\ 0 \le \epsilon_1, \epsilon_2 \le 1\} \ . \end{aligned}$$

Thus the distorted density $p(\mathbf{X}|W)$ (and $P(W)$) is a mixture of the hypothetical distribution $p_{\Lambda_0}(\mathbf{X}|W)$ (and $P_{\Gamma_0}(W)$), and an arbitrary distribution $h_1(\mathbf{X}|W)$ (and $h_2(W)$) describing the possible distortions. This type of distortion model is the most popular one in robust statistics [47, 41]. How to derive the relevant robust decision rule under this distortion model remains an interesting problem for future research.

3.6.4 Discussion

The crucial difference between the plug-in and predictive classifiers is that the former acts as if the estimated model parameters were the true ones, whereas, the predictive methods average over the uncertainty in parameters. However, if we use the posterior PDF $p(\Lambda, \Gamma|\mathcal{X})$ in Eq.(3.31) derived from the *training set* $\mathcal{X}$ directly to serve as the prior PDF in predictive decision making, BPC will make little difference from the conventional plug-in MAP rule in many applications. This is because whatever initial prior PDF, $p(\Lambda, \Gamma|\varphi_\Lambda^{(0)}, \varphi_\Gamma^{(0)})$, is used, when a large amount of training data $\mathcal{X}$ are available, we will get a posterior PDF $p(\Lambda, \Gamma|\mathcal{X})$ with a sharp peak. This makes the predictive PDFs in Eqs. (3.32) and (3.33) of little difference from $p_{\hat{\Lambda}}(\mathbf{X}|W)$ and $P_{\hat{\Gamma}}(W)$ with the ML estimates $\hat{\Lambda}, \hat{\Gamma}$. In the limit, if the posterior probability mass of (Λ, Γ) is concentrated at the ML estimate $(\hat{\Lambda}, \hat{\Gamma})$ obtained from $\mathcal{X}$, it is easy to see from Eqs. (3.34), (3.32), and (3.33) that the BPC decision rule coincides with the plug-in MAP decision rule.

Historically, the predictive classification approach receives little attention in many classical statistics textbooks despite the existence of many good works [4, 30, 31]. As pointed out by Ripley [106], this may be because it usually makes little difference from plug-in approaches within the problems and the tightly-constrained parametric families many statisticians use or consider. Nonetheless, it will become important when we consider much larger families and formulate the problem appropriately as shown before. To our knowledge, it was Nadas who first

- adopted a BPC formulation and pointed out its potential in speech recognition applications [98]; and
- explicitly stated and proved the optimality of BPC in the sense of minimizing overall risk (Ripley also discussed the predictive classification approach in this way in his recent pattern recognition textbook [106]).

However, like other statisticians, Nadas was directly using the posterior PDF $p(\Lambda, \Gamma|\mathcal{X})$ to serve as the prior PDF in predictive decision making and gave a simple example in which a *reproducing density* exists. No experimental results were reported and the paper closed by briefly discussing the difficulty of applying the theory to HMM-based speech recognition.

Starting from Nadas's formulation, Merhav and Ephraim [94] suggested a so-called *approximate Bayesian (AB) decision rule* for speech recognition which was based on the generalized likelihood ratios computed from the available training and testing data. Such an AB rule operates as follows:

$$\hat{W} = \underset{W}{\operatorname{argmax}} \frac{\max_\Lambda [p(\mathbf{X}|\Lambda, W) \cdot p(\mathcal{X}|\Lambda, W)]}{\max_\Lambda p(\mathcal{X}|\Lambda, W)} P_{\Gamma_0}(W) \quad . \tag{3.35}$$

It is clear that if the training sequences $\mathcal{X}$ are considerably longer than the test sequence $\mathbf{X}$ which is the case in most speech recognition applications, the parameter set Λ that maximizes the denominator of Eq. (3.35) is very close to the parameter

set that maximizes the numerator; hence the factor $p(\mathcal{X}|\Lambda, W)$ in both the numerator and denominator is essentially canceled. This makes the AB decision rule of little difference from the plug-in MAP decision rule using an ML estimate of Λ. The AB decision rule is also computationally expensive because the maximization of $[p(\mathbf{X}|\Lambda, W) \cdot p(\mathcal{X}|\Lambda, W)]$ over Λ must be performed for every test sequence $\mathbf{X}$. Furthermore, all of the training data must be stored. All of these factors make the AB decision rule impractical for most speech recognition applications.

As discussed previously, the minimax classification rule can be viewed as a two-step procedure and implemented in Eq. (3.29). First, each testing utterance is treated as possibly belonging to any word sequence and a constrained ML estimate of the related parameters is obtained. Then, a plug-in MAP rule is used for speech recognition by using the updated parameters. This intuitive interpretation opens up the possibilities to use other estimation approaches, e.g. the MAP approach, in the first step. Such a modified minimax decision rule works as follows:

$$\hat{W} = \underset{W}{\operatorname{argmax}}[p(X|\Lambda_{MAP}, W) \cdot P_{\Gamma_0}(W)] \ , \tag{3.36}$$

where Λ_{MAP} is the MAP estimate of Λ. For the convenience of reference, we call this modified minimax decision rule as a *Bayesian minimax rule* to emphasize its difference from the minimax approach in [95].

We have previously discussed the BPC approach as a new decision rule which averages out the sampling error in parameter estimation. A related but simpler approach can also be used. For example, for a CDHMM-based ASR system, instead of directly modifying the basic decision rule, one can also assume that the CDHMM parameters are uncertain. Then one uses the *Bayesian predictive density* of each Gaussian mixture component to serve as the compensated distribution of that component and plug these compensated distributions into the plug-in MAP decision rule in Eq. (3.16). The approach is thus called *Bayesian predictive density based model compensation* method, or shortly BP-MC method, to differentiate it from the BPC rule defined in Eq. (3.34). In [110], such an idea is explored in the context of Bayesian speaker adaptation where a Gaussian prior PDF for mean vector is adopted. In [60], a similar idea is applied to noisy speech recognition where a uniform prior PDF on a pre-specified uncertainty neighborhood for mean vector is adopted. More recently, similar ideas are applied to the transformation-based model compensation by using the predictive PDF of the transformation parameters [114, 111, 14, 13].

3.7 Summary

In this chapter, we have revisited the decision theoretic foundation of the modern ASR technology. We have explained several key concepts about the optimal decision rule, adaptive decision rule, and robust decision rule. We have shown how these decision rules can be derived under different assumptions and optimality criteria. A

clear understanding of these assumptions and criteria will guide us to appreciate why the current ASR technology is so successful in certain applications and more importantly why it fails in many other situations. Consequently, we are able to discuss the rationale of several ways of improving adaptive decision rules via decision parameter adaptation. Most of the discussions in this chapter can also be applied to other pattern recognition problems employing the same decision-theoretic formulation.

Before closing the chapter, we do want to point out one important unsatisfactory fact. We are treating each W as a *class* which could mean different things as explained at the beginning of Section 3.2. In the case of continuous speech recognition, W takes the form of a sequence of other smaller linguistic units such as words in normal sense. All of the decision rules described in this chapter aim at achieving the minimum classification error of W instead of the word recognition error rate, which is usually used as a measure of practical ASR performance. Apparently there is a mismatch here too! Unfortunately, the method to derive a decision rule which achieves the minimum word recognition error rate (possibly by using a loss-function beyond "0-1") remains an interesting open problem. Readers are referred to some interesting recent works that try to attack this problem [83, 77, 34, 119].

It is our hope that the in-depth discussions in this chapter may inspire further innovations that will lead to better solutions for ASR and many other pattern recognition applications.

Acknowledgement

The author gratefully acknowledges the contributions of his past and present collaborators, including C. Chan, C.-H. Lee, H. Jiang, B. Ma, and J. Wu. This work was funded by grants from the RGC of the Hong Kong SAR (Project Numbers HKU7022/00E and HKU7039/02E).

References

[1] Acero, A., *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, 1993.

[2] Afify, M., and O. Siohan, "Sequential noise estimation with optimal forgetting for robust speech recognition," *Proc. of ICASSP-2001*, 2001.

[3] Afify M., O. Siohan, and C.-H. Lee, "Upper and lower bounds on the mean of noisy speech: application to minimax classification," *IEEE Trans. on Speech and Audio Processing*, Vol. 10, No. 2, pp.79-88, 2002.

[4] Aitchison, J. and I. R. Dunsmore, *Statistical Prediction Analysis*, Cambridge, UK: Cambridge University Press, 1975.

[5] Amari, S., "A theory of adaptive pattern classifiers," *IEEE Trans. on Electronic Computers*, Vol. EC-16, No. 3, pp.299-307, 1967.

[6] Bahl, L. R., F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 2, pp.179-190, March 1983.

[7] Bahl, L. R., P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. of ICASSP-86*, 1986, pp.49-52.

[8] Bahl, L. R., P. F. Brown, P. V. de Souza and R. L. Mercer, "Estimating hidden Markov model parameters so as to maximize speech recognition accuracy," *IEEE Trans. Speech and Audio Processing*, Vol. 1, No. 1, pp.77-83, 1993.

[9] Baker, J. K., "Stochastic modeling for automatic speech understanding," in *Speech Recognition* (D. R. Reddy, ed.), New York: Academic, 1975, pp.521-542.

[10] Baker, J. K., "The DRAGON system – an overview," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, pp.24-29, 1975.

[11] Baum, L. E., "An inequality and associated maximization techniques in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, Vol. 3, pp.1-8, 1972.

[12] Chengalvarayan, R., "Speaker adaptation using discriminative linear regression on time-varying mean parameters in trended HMM," *IEEE Signal Processing Letters*, Vol. 5, No. 3, pp. 63-65, 1998

[13] Chien, J.-T., "Combined linear regression adaptation and Bayesian predictive classification for robust speech recognition," *Proc. of Eurospeech-2001*, Aalborg, Denmark, Sept. 2001.

[14] Chien, J.-T. and G.-H. Liao, "Transformation-based Bayesian predictive classification using online prior evolution," *IEEE Trans. on Speech and Audio Processing*, Vol. 9, No. 4, pp.399-410, 2001.

[15] Chou, W., "Discriminant-function-based minimum recognition error rate pattern-recognition approach to speech recognition," *Proceedings of the IEEE*, Vol. 88, No. 8, pp.1201-1223, 2000.

[16] Delphin-Poulat, L., C. Mokbel, and J. Idier, "Frame-synchronous stochastic matching based on the Kullback-Leibler information," *Proc. of ICASSP-1998*, 1998, pp.89-92.

[17] Deng, L., "A dynamic feature based approach to the interface between phonology and phonetics for speech modeling and recognition," *Speech Communication*, Vol. 24, No. 4, pp. 299-323, 1998.

[18] Duda, R. O. and Hart, P. E., *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

[19] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*, 2nd ed., New York: Wiley, 2001.

[20] De Mori, R. (ed.), *Spoken Dialogues with Computers*, Academic Press, 1998.

[21] Ephraim, Y., A. Dembo, and L. R. Rabiner, "A minimum discrimination information approach for hidden Markov modeling," *IEEE Trans. on Information Theory*, Vol. 35, No. 5, pp.1001-1013, 1989.

[22] Ephraim, Y. and L. R. Rabiner, "On the relations between modeling approaches for speech recognition," *IEEE Trans. on Information Theory*, Vol. 36, No. 2, pp.372-380, 1990.

[23] Ephraim, Y., "Statistical model based speech enhancement systems," *Proc. IEEE*, Vol. 80, No. 10, pp. 1526-1555, 1992.

[24] Ferguson, T. S., *Mathematical Statistics: a Decision Theoretic Approach*, New York: Academic Press, 1967.

[25] Furui, S., "Recent advances in robust speech recognition," *Proc. ETRW on Robust Speech Recognition for Unknown Communication Channels*, Pont-a-Mousson, France, April 1997, pp.11-20.

[26] Gales, M. J. F., "Predictive model-based compensation schemes for robust speech recognition," *Speech Communication*, Vol. 25, pp.47-74, 1998.

[27] Gales, M. J. F., "Cluster adaptive training of hidden Markov models," *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 4, pp.417-428, 2000.

[28] Gao, Y.-Q., M. Padmanabhan, and M. A. Picheny, "Speaker adaptation based on pre-clustering training speakers," *Proc. of Eurospeech-97* (Rhodes, Greece), 1997, pp.2091-2094.

[29] Gao, Y.-Q., Y.-X. Li and M. Picheny, "Maximal rank likelihood as an optimization function for speech recognition," *Proc. ICSLP-00*, Beijing, Oct. 2000.

[30] Geisser, S., "Bayesian Discrimination", in *Handbook of Statistics*, P. R. Krishnaiah and L. N. Kanal, eds., Vol. 2, pp.101-120, 1982.

[31] Geisser, S., *Predictive Inference: An Introduction*, New York: Chapman & Hall, 1993.

[32] Glick, N., "Sample-based classification procedures derived from density estimators," *Journal of the American Statistical Association*, Vol. 67, pp.116-122, 1972.

[33] Glick, N., "Sample-based classification procedures related to empiric distributions," *IEEE Trans. on Information Theory*, Vol. 22, pp.454-461, 1976.

[34] Goel, V. and W. Byrne, "Minimum Bayes-risk automatic speech recognition," *Computer Speech and Language*, Vol. 14, pp.115-135, 2000.

[35] Gong, Y., "Speech recognition in noisy environments: a survey", *Speech Communication*, Vol. 16, pp.261-291, 1995.

[36] Gong, Y., "Stochastic trajectory modeling and sentence searching for continuous speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 5, No. 1, pp.33-44, 1997.

[37] Good, I. J., "The population frequencies of species and the estimation of population parameters," *Biometrika*, Vol. 40, pp. 237-264, 1953.

[38] Gopalakrishnan, P. S., D. Kanevsky, A. Nadas, D. Nahamoo, and M. A. Picheny, "Decoder selection based on cross-entropies," in *Proc. ICASSP-88*, 1988, pp.20-23.

[39] Gotoh, Y., Hochberg, M. M., and Silverman, H. F., "Efficient training algorithms for HMMs using incremental estimation," *IEEE Trans. on Speech and Audio Processing*, Vol. 6, No. 6, pp.539-548, 1998.

[40] Gunawardana A. and W. Byrne, "Discriminative speaker adaptation with conditional maximum likelihood linear regression," *Proc. Eurospeech-01*, Aalborg, Denmark, 2001.

[41] Hample, F. R., E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The approach Based on Influence Functions*, New York: John Wiley & Sons, 1986.

[42] Hazen, T. J., "A comparison of novel techniques for rapid speaker adaptation," *Speech Communication*, Vol. 31, pp.15-33, 2000.

[43] Hermansky, H., "Should recognizers have ears?," *Speech Communication*, Vol. 25, pp.3-27, 1998.

[44] Ho, Y. C. and Agrawala, A. K., "On pattern classification algorithms – introduction and survey," *IEEE Trans. on Automatic Control*, Vol. AC-13, pp.676-690, 1968.

[45] Huang, J., and M. Padmanabhan, "A study of adaptation techniques on a voicemail transcription task," *Proc. of Eurospeech-99*, 1999.

[46] Huang, X.-D., Acero A., and Hon, H. W., *Spoken language processing: a guide to theory, algorithm, and system development*, Upper Saddle River, N.J.: Prentice Hall, 2001.

[47] Huber, P. J., *Robust Statistics*, New York: John Wiley & Sons, 1981.

[48] Huo, Q. and C.-H. Lee, "On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate," *IEEE Trans. on Speech and Audio Processing*, Vol. 5, No. 2, pp.161-172, 1997.

[49] Huo, Q. and C.-H. Lee, "On-line adaptive learning of the correlated continuous density hidden Markov models for speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 6, No. 4, pp.386-397, 1998.

[50] Huo Q. and C.-H. Lee, "A Bayesian predictive classification approach to robust speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 2, pp.200-204, 2000.

[51] Huo, Q. and Lee, C.-H., "Robust speech recognition based on adaptive classification and decision strategies," *Speech Communication*, Vol. 34, pp.175-194, 2001.

[52] Huo, Q. and B. Ma, "Robust speech recognition based on off-line elicitation of multiple priors and on-line adaptive prior fusion," *Proc. ICSLP-2000*, Beijing, China, October 2000, pp.IV-480-483.

[53] Huo, Q. and B. Ma, "On-line adaptive learning of continuous density hidden Markov models based on multiple-stream prior evolution and posterior pooling," *IEEE Trans. on Speech and Audio Processing*, Vol. 9, No. 4, pp.388-398, May 2001.

[54] Huo, Q., Smith N., and Ma, B., "Efficient ML training of CDHMM parameters based on prior evolution, posterior intervention and feedback," *Proc. ICASSP-2000*, Turkey, 2000, pp.II-1001-1004.

[55] Jain, A. K., Duin, R. P. W., and Mao, J., "Statistical pattern recognition: a review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, pp.4-37, 2000.

[56] Jelinek, F., "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, Vol. 64, No. 4, pp.532-556, April 1976.

[57] Jelinek, F., *Statistical Method for Speech Recognition*, The MIT Press, Cambridge, 1997.

[58] Jelinek, F., R. L. Mercer, and S. Roukos, "Principles of lexical language modeling for speech recognition," in *Advances in Speech Signal Processing*, S. Furui and M. M. Sondhi (eds.), New York: Marcel Dekker, 1991, pp.651-699.

[59] Jiang, H. and L. Deng, "A robust compensation strategy for extraneous acoustic variations in spontaneous speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 10, No. 1, pp.9-17, 2002.

[60] Jiang, H., K. Hirose, and Q. Huo, "Robust speech recognition based on a Bayesian prediction approach," *IEEE Trans. on Speech and Audio Processing*, Vol. 7, No.4, pp. 426-440, 1999.

[61] Jiang, H., K. Hirose, Q. Huo, "Improving Viterbi Bayesian predictive classification via sequential Bayesian learning in robust speech Recognition," *Speech Communication* Vol. 28, No. 4, pp. 313-326, 1999.

[62] Jiang, H., K. Hirose, Q. Huo, "A minimax search algorithm for robust continuous speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 6, pp.688-694, 2000.

[63] Juang, B.-H., S. E. Levinson and M. M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains," *IEEE Trans. on Information Theory*, Vol. IT-32, No. 2, pp.307-309, 1986.

[64] Juang, B.-H. and L. R. Rabiner, "The segmental k-means algorithm for estimating parameters of hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 9, pp.1639-1641, 1990.

[65] Juang, B.-H. and L. R. Rabiner, "Hidden Markov models for speech recognition," *Technometrics*, Vol. 33, No. 3, pp.251-272, 1991.

[66] Juang, B.-H., "Speech recognition in adverse environments," *Computer Speech and Language*, Vol. 5, pp.275-294, 1991.

[67] Juang, B.-H. and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. on Signal Processing*, Vol. 40, No. 12, pp.3043-3054, 1992.

[68] Juang, B.-H., "Automatic Speech Recognition: Problems, Progress & Prospects," Handout of Keynote Speech, *1996 IEEE Workshop on Neural Networks For Signal Processing*, Kyoto, 1996.

[69] Juang, B.-H., W. Chou and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 5, No. 3, pp.257-265, 1997.

[70] Junqua, J.-C. and Haton, J.-P., *Robustness in Automatic Speech Recognition: Fundamentals and Applications*, Kluwer Academic Publishers, Boston, 1996.

[71] Kanal, L., "Patterns in pattern recognition: 1968-1974," *IEEE Trans. on Information Theory*, Vol. IT-20, pp.697-722, 1974.

[72] Katagiri, S., B.-H. Juang, and C.-H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proc. of IEEE*, Vol. 86, No. 11, pp.2345-2373, 1998

[73] Katz, S. M., "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 35, No. 3, pp. 400-401, 1987.

[74] Kosaka, T., S. Matsunaga, and S. Sagayama, "Speaker-independent speech recognition based on tree-structured speaker clustering," *Computer Speech and Language*, Vol. 10, pp.55-74, 1996.

[75] Kuhn, R., J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 6, pp.695-707, 2000.

[76] Mangu, L., E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, Vol. 14, pp.373-400, 2000.

[77] Mangu, L. and M. Padmanabhan, "Error corrective mechanisms for speech recognition," *Proc. of ICASSP-2001*, 2001.

[78] Kermorvant, C. and C. Mokbel, "Towards introducing long-term statistics in MUSE for robust speech recognition," *Proc. of ASRU-1999*, 1999.

[79] Kharin, Y., *Robustness in Statistical Pattern Recognition*, Boston: Kluwer Academic, 1996.

[80] Kim, N.-S., "Nonstationary environment compensation based on sequential estimation," *IEEE Signal Processing Letters*, Vol. 5, No. 3, pp.57-59, 1998.

[81] Kim, N.-S., "IMM-based estimation for slowly evolving environments," *IEEE Signal Processing Letters*, Vol. 5, No. 6, pp.146-149, 1998.

[82] Korkmazskiy F. and B.-H. Juang, "Discriminative adaptation for speaker verification," *Proc. ICSLP-96*, 1996.

[83] Laurila, K., M. Vasilache, and O. Viikki, "A combination of discriminative and maximum likelihood techniques for noise robust speech recognition," *Proc. ICASSP-98*, 1998, pp.85-88.

[84] Lee, C.-H., F.-K. Soong and K.-K. Paliwal (eds.), *Automatic Speech and Speaker Recognition: Advanced Topics*, Boston: Kluwer Academic Publishers, 1996.

[85] Lee, C.-H., "On stochastic feature and model compensation approaches to robust speech recognition," *Speech Communication*, Vol. 25, pp. 29-47, 1998.

[86] Lee, C.-H. and Huo Q., "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," *Proceedings of the IEEE*, Vol. 88, No. 8, pp.1241-1269, 2000.

[87] Levinson, S. E., Rabiner L. R., and Sondhi, M. M., "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *The Bell System Technical Journal*, Vol. 62, No.4, pp.1035-1074, 1983.

[88] Levinson, S. E., "Structural Methods in Automatic Speech Recognition," *Proc. IEEE*, Vol. 73, pp. 1625-1650, 1985.

[89] Lin, C.-H., C.-H. Wu, and P.-C. Chang, "A study on speaker adaptation for Mandarin syllable recognition with minimum error discriminative training", *IEICE Trans. Inf. & Syst.* , Vol. E78-D, No. 6, pp.712-718, 1995.

[90] Lee, K.-F., *Automatic Speech Recognition – The Development of the SPHINX-System*, Kluwer Academic Publishers, Boston, 1989.

[91] Liporace, L. R., "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. on Information Theory*, Vol. IT-28, pp.729-734, 1982.

[92] Ljolje, A., Y. Ephraim, and L. R. Rabiner, "Estimation of hidden Markov model parameters by minimizing empirical error rate," in *Proc. ICASSP-90*, 1990, pp.709-712.

[93] Matsui, T. and S. Furui, "A study of speaker adaptation based on minimum classification error training," *Proc. Eurospeech-95*, Madrid, September 1995, pp.81-84.

[94] Merhav, N. and Y. Ephraim, "A Bayesian classification approach with application to speech recognition," *IEEE Trans. on Signal Processing*, Vol. 39, No. 10, pp.2157-2166, 1991.

[95] Merhav, N. and C.-H. Lee, "A minimax classification approach with application to robust speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 1, No. 1, pp.90-100, 1993.

[96] Moon, S. and J.-N. Hwang, "Robust speech recognition based on joint model and feature space optimization of hidden Markov models," *IEEE Trans. on Neural Networks*, Vol. 8, No. 2, pp.194-204, 1997.

[97] Nadas, A., "A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, No. 4, pp.814-817, 1983.

[98] Nadas, A., "Optimal solution of a training problem in speech recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, No. 1, pp.326-329, 1985.

[99] Nadas, A., D. Nahamoo, and M. A. Picheny, "On a model-robust training method for speech recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-36, No. 9, pp.1432-1436, 1988.

[100] Nagy, G., "State of the art in pattern recognition," *Proceedings of the IEEE*, Vol. 56, pp.836-862, 1968.

[101] Ney, H. and Ortmanns S., "Progress in dynamic programming search for LVCSR," *Proceedings of the IEEE*, Vol. 88, No. 8, pp.1224-1240, 2000.

[102] Ostendorf, M., V. V. Digalakis, and O. A. Kimball, "From HMM's to segment models: a unified view of stochastic modeling for speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 4, No. 5, pp.360-378, 1996.

[103] Rabiner, L. R., J. G. Wilpon and B.-H. Juang, "A segmental K-means training procedure for connected word recognition," *AT&T Tech. Journal*, Vol. 65, pp. 21-31, 1986.

[104] Rabiner, L. R., "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, pp.257-286, 1989.

[105] Rabiner, L. R. and Juang, B.-H., *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[106] Ripley, B. D., *Pattern Recognition and Neural Networks*, Cambridge, UK: Cambridge University Press, 1996.

[107] Robbins, H. and H. Monroe, "A stochastic approximation method," *Annals of Mathematical Statistics*, Vol. 22, pp.400-407, 1951.

[108] Russell, M. J. and Holmes, W. J., "Linear trajectory segmental HMM's," *IEEE Signal Processing Letters*, Vol.4, No.3, pp.72-74, 1997.

[109] Sankar, A. and C.-H. Lee, "A maximum likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 4, No. 3, pp.190-202, 1996.

[110] Shahshahani, B. M., "A Markov random field approach to Bayesian speaker adaptation," *IEEE Trans. on Speech and Audio Processing*, Vol. 5, No. 2, pp. 183-191, 1997.

[111] Siohan, O. and A. C. Surendran, "Structural Bayesian predictive adaptation of hidden Markov models," *Proc. Workshop on Adaptation Methods for Speech Recognition*, Sophia-Antipolis, France, August 2001.

[112] Stern, R. M., Raj, B., and Moreno, P. J., "Compensation for environmental degradation in automatic speech recognition," *Proc. ETRW on Robust Speech Recognition For Unknown Communication Channels*, Pont-a-Mousson, France, April 1997, pp.33-42.

[113] Surendran, A. C., C.-H. Lee, and M. Rahim, "Non-linear compensation for stochastic matching," *IEEE Trans. on Audio and Speech Processing*, Vol. 7, No. 6, pp. 643-655, 1999.

[114] Surendran, A. C. and C.-H. Lee, "Transformation-based Bayesian prediction for adaptation of HMMs," *Speech Communication*, Vol. 34, pp.159-174, 2001.

[115] Takahashi J. and S. Sagayama, "Discriminative training based on minimum classification error for a small amount of data enhanced by vector-field-smoothed Bayesian learning," *IEICE Trans. Inf. & Syst.*, Vol. E79-D, No. 12, pp.1700-1707, 1996.

[116] Tsypkin, Y. Z., *Adaptation and learning in automatic systems*, New York: Academic Press, 1971.

[117] Tsypkin, Y. Z., *Foundations of the theory of learning systems*, New York: Academic Press, 1973.

[118] Uebel, L. and P. C. Woodland, "Improvements in linear transform based speaker adaptation," *Proc. ICASSP-01*, 2001.

[119] Wessel, F., R. Schluter, and H. Ney, "Explicit word error minimization using word hypothesis posterior probabilities," *Proc. of ICASSP-2001*, 2001.

[120] Wallhoff, R., D. Willett, and G. Rigoll, "Frame-discriminative and confidence-driven adaptation for LVCSR", *Proc. ICASSP-00*, 2000.

[121] Wald, A., *Statistical Decision Functions*, New York: Wiley, 1950.

[122] Wang, Z., and F. Liu, "Speaker adaptation using maximum likelihood model interpolation," *Proc. of ICASSP-99*, 1999.

[123] Wu, J. and Q. Huo, "Supervised adaptation of MCE-trained CDHMMs using minimum classification error linear regression," *Proc. ICASSP-2002*, Orlando, Florida, May 2002.

[124] Yao, K., K. K. Paliwal, and S. Nakamura, "Sequential noise compensation by a sequential Kullback proximal algorithm," *Proc. of Eurospeech-2001*, 2001.

[125] Young, S, Kershaw, D. J., Odell, J. J., Ollason, D., Valtchev V., and Woodland, P. C., *The HTK Book Version 3.0*, Cambridge University, Cambridge, England, 2000.

4

Speech Pattern Recognition using Neural Networks

Shigeru Katagiri
NTT Communication Science Laboratories

CONTENTS

4.1 Introduction

Traditionally, the development of speech pattern recognition systems has been attempted by using the pattern matching technology based on distance computation incorporating dynamic programming (DP). In this approach, an input speech pattern is represented as a sequence of acoustic feature vectors, compared with class models, each represented in the same manner as the input pattern, and then decoded to the model class closest to the input, in terms of DP-based distance. This simple scheme was practical and effective for implementing recognizers in the then limited computational environment, and indeed many noteworthy systems were developed for connected word recognition as well as basic isolated word recognition (e.g., see [22]).

In the 1980's, there were two epochs in speech pattern recognition research. One was a paradigm shift from pattern matching to a new, probabilistic classification decision paradigm, which mainly relied on the use of hidden Markov models (HMMs). The other was the challenge of employing a rapidly growing technological paradigm called neural networks (NNs) (e.g., see [16, 24]). In these new paradigmatic stages, speech patterns were basically represented in the same fashion as in the classic pattern matching. However, in these new stages, classes to which an input should be decoded were modeled by more elaborated structures such as HMMs and NNs.

The probabilistic decision framework is suitable for efficiently modeling the statis-

tical variation of speech samples. It is also useful for unifying different types of information sources, such as acoustic speech models and linguistic speech models. Indeed, the use of HMM has greatly contributed to advancing speech recognition technologies, and HMM is still a mainstream choice as a recognizer structure. The use of HMM is probably just one of many factors contributing to recent advances. However, it should be also noted that most of the current recognizers successfully employ HMMs in commercial services that even encompass large-vocabulary, speaker-independent, connected-word, and acoustically corrupted telephone-based recognition tasks.

NN is characterized by its high discriminative capability. Actually, since the advent of the historic breakthrough of the multilayer perceptron (MLP) network, NN has been extensively applied to speech recognition in various forms, such as thoroughbred NN-based recognizers and hybrids of NN and HMM. Through vigorous studies, it was shown that the NN's high classification capability mainly originated from its discriminative training methods, such as using the minimization of the squared error loss or using the minimization of the classification error count loss. Due to this special feature for training, NN is now established as a modern class of design methodology for achieving high-performance speech pattern recognizers.

As its name implies, an NN-based system usually consists of a network structure and many simple operation units. NN thus essentially possesses a distributed and parallel computation mechanism that is advantageous in speed and robustness of the operation. Specifically, there are two types of robustness: 1) one that increases the fault tolerance of operation, of which degradation is due to hardware troubles, and 2) one that increases the stability/reliability of a classification decision, which is a long-standing mathematical research issue in decision theories such as the Bayes decision theory. The latter is clearly an important algorithm-related issue that should be vigorously studied in the research field of NN-based pattern recognition. Among the many candidate solutions for improving decision reliability, there has been rigorous investigation of combining multiple decisions by decision fusion to obtain a final reliable classification output.

Many textbooks and handbooks have already comprehensively tracked the development history of NN-based speech pattern recognizers (e.g., see [16]). However, these are not necessarily useful for studying actual advantages/disadvantages, which are primarily based on the selection of training methods, of the NN-based speech recognizers. In light of this, we introduce in this chapter NN-based speech recognition attempts with a special focus on their training procedures. As cited above, the procedures are basically categorized as discriminative training originating in the classic linear discriminant analysis, which is still a mainstream recognizer design in the most fundamental theoretical framework of pattern recognition, i.e., the Bayes decision theory (e.g., see [4]). As a basis for comprehensively covering discriminative training issues, we use a recent general discriminative training formalism called the generalized probabilistic descent (GPD) method [10, 11, 12, 14, 15, 17].

The chapter is organized as follows. After the present section, we provide in Section 4.2 the fundamentals of the Bayes decision theory. In this overview, we use a novel GPD-based description. In Section 4.3, we discuss NN-based speech recognizers,

with an emphasis on their training procedures. In Section 4.4, we discuss the robustness of NN-based recognition, paying attention to the issue of decision fusion. Finally, we provide concluding remarks in Section 4.5.

4.2 Bayes Decision Theory

4.2.1 Preparations

For discussion purposes, we use an exemplar task setting of recognizing a speech pattern U with a modular recognizer composed of a front-end feature extraction module (feature extractor) and a back-end classification module (classifier). The number of classes to which U may belong is M. The recognizer contains a set of trainable parameters Ψ ($= \{\Phi, \Lambda\}$), where Φ is the set of trainable parameters of the feature extractor, and Λ is for the same classifier.

Speech patterns are essentially dynamic, i.e., of a variable length and a nonlinearly warping temporal structure. To maintain its dynamic nature, a speech pattern is usually represented as a sequence of acoustic feature vectors, each calculated by shifting a short-time window over the observed pattern. For our task setting, we assume that at the feature extractor U is converted to vector sequence $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T\}$, where $\mathbf{x}_t$ is the F-dimensional acoustic feature vector at time index t over U and F is a fixed number. The task of the classifier is then to decode X to its corresponding correct class C_i, which is one of the M possible classes of C_1 through C_M.

4.2.2 Decision Rule

Intuition suggests that a natural rule of a classification decision is to classify an input into its most likely class. A discriminant function is introduced to measure this likelihood.

For simplicity, let us assume that the feature extractor is determined independently of the classifier's design. Given the feature extraction operation, we formally define this classification operation as follows:

$$C(\mathbf{x}) = C_i \qquad \text{iff } i = \arg\max_j g_j(X; \Lambda), \tag{4.1}$$

where $C(\mathbf{x})$ is the classification operation, and $g_j(X; \Lambda)$ is the discriminant function of C_j that measures the degree to which X belongs to C_j. It turns out here that an immediate goal of recognizer design is to train Λ so that the classifier can decode X correctly. However, the recognizer is naturally expected to handle many input patterns. The ultimate design goal thus becomes achieving the status of Λ that leads to the most accurate classification over the entire set of available input speech patterns.

4.2.3 Minimum Error-rate Classification

Rule (4.1) and the term of likelihood imply that we should use a probability function as a discriminant function. Then, assuming that the probability for dynamic patterns is defined properly, (4.1) becomes the following quite natural rule:

$$C(X) = C_i \qquad \text{iff } i = \arg\max_j p(C_j|X), \tag{4.2}$$

which requires X to be classified as the class having the largest a posteriori probability.

The decision using (4.2) leads to the optimal minimum error-rate classification, or, in other words, the minimum classification error probability condition, which is an optimal design goal in the statistics-based theory (e.g., see [4]).

4.2.4 Probability Function Estimation

If one can accurately estimate $p(C_j|X)$ with $g_j(X;\Lambda)$, one can in principle achieve the optimal minimum error-rate classification. It should be noted, however, that the a posteriori probability is only computed over an infinite set of samples, and therefore such an achievement using (4.2) is nearly impossible in practical design problems.

Nevertheless, many attempts have been made to accurately estimate the probabilities based on the sound and tractable mathematical bases of the maximum likelihood estimation (MLE) method and the Bayesian estimation method.

In these methods, (4.2) is usually rewritten as

$$C(X) = C_i \qquad \text{iff } i = \arg\max_j p(X|C_j)p(C_j), \tag{4.3}$$

where $p(C_j|X)$ is replaced by the class-conditional probability $p(X|C_j)$ and the a priori probability $p(C_j)$. These probability functions are more suitable to the use of the MLE and Bayesian methods.

Based on the accumulated research results, the approach using the probability function estimation has made important advances. However, in reality, it still suffers from difficult problems and there are many ongoing research efforts. Fundamental open questions are summarized as follows. First, functional forms, such as a Gaussian function, of the probability functions are rarely known, and therefore the mismatch between a true functional form and its counterpart form selected for estimation causes unavoidable estimation errors. Second, the approach assumes that a large number of design samples are available for estimation, while it is often difficult to collect a sufficient number of design samples. The estimation process is essentially based on the matching between a selected-form probability function and the sample distribution, and thus the approach inevitably relies on the above assumption. It then becomes an intractable mystery how corrupted the probability function estimates are over a limited number of design samples. The last question is how the corrupted estimates affect the performance of (4.3). The optimality of (4.3) holds only in the case of error-free estimation of the probabilities. Basically, the more accurate the

probability estimates are, the more accurate the classification decision using (4.3) is. However, the estimation improvement can occur only individually on the probability functions, each of a different class, and no clear perspective has been available on the interaction in classification accuracy between the probability estimation and the classification decision. It should be noted here that the method using the probability function estimates is an indirect approach to improving the performance of the entire decision rule, and the indirectness is a fundamental cause of the last question.

4.2.5 Discriminative Training

A more direct approach than the MLE and Bayesian methods to a successful execution of the classification decision returns to (4.1). The approach originated in the classic training method called linear discriminant analysis, and it is usually referred to as discriminative training. It emulates the entire classification (discrimination) operation of (4.1) and attempts to realize a set of discriminant functions (consequently Λ) that achieve the desirable, in terms of a preset training objective (criterion), classification performance for training samples.
The key design issues that determine the performance of discriminative training include the following: 1) how to define the discriminant functions, 2) how to evaluate the performance of decisions in the training stage, 3) how to adjust trainable system parameters (e.g. Λ), and 4) how to cope with unknown samples that do not appear in the training stage. It turns out that these points comprehensively cover the classifier design procedure. These issues will be described in turn in later pages.

4.2.5.1 Functional Form Embodiment of the Entire Process

Emulation of the classification process inevitably requires that the process be represented in a tractable functional form, which enables one to deal with the design of its corresponding classification system mathematically. Based on an observation of the decision process in (4.1), one can see that the decision consists of the comparison of the discriminant functions over all possible classes. A general way of emulating the process is given in the following functional form, which is often referred to as a misclassification measure:

$$d_k(X;\Lambda) = -g_k(X;\Lambda) + \left[\frac{1}{M-1} \sum_{j,j\neq k} \{g_j(X;\Lambda)\}^{\mu} \right]^{1/\mu}, \qquad (4.4)$$

where μ is a positive constant. One can notice here that given X of C_k, $d_k(\,) > 0$ indicates a misclassification, and $d_k(\,) < 0$ indicates a correct classification. In addition, controlling μ enables the simulation of various decision rules. In particular, when μ approaches ∞, (4.4) closely emulates rule (4.1).

4.2.5.2 Discriminant Functions

The selection of a functional form of the discriminant function is basically equivalent to the selection of a measure (or measurement) used to represent the degree to which

an input sample belongs to some class. In addition, the form relies on the type of trainable system parameters, i.e. Λ, and it is generally selected based on the nature of the patterns.

Typical examples of classical forms (for fixed-dimensional patterns) include the linear discriminant function and the distance. The measure used in the linear discriminant function is a linearly-weighted sum of input vector components, where a set of weights corresponds to trainable system parameters Λ. In the distance case, the measure is a reasonably selected distance between an input and a reference vector, such as Euclidean distance, where the reference vectors work as Λ.

For the case of classifying dynamic patterns, such as speech samples, system parameters should contain some means of representing the temporal structure. A traditional selection employs reference vectors and uses a DP-based normalized distance measure between an input pattern and a reference vector sequence. On the other hand, in most recent cases, trainable system parameters are the component probabilities of HMM, such as state transition probability and observation (emission) probability. The temporal structure is represented therein as the state transition structure.

In the cases using NNs, a discriminant function is basically an output of NN's output node. The type of measure therein relies on the selection of node functions, such as perceptron-like activation function, radial-basis activation function, and sigmoidal output function.

4.2.5.3 Loss over an Individual Pattern

To evaluate decision performance in the design stage, individual loss is introduced for every training pattern, which is a function of the misclassification measure and reflects classification degradation. It should be noted that (individual) loss is sometimes referred to as (individual) risk or (individual) design objective. Obviously, the smaller the loss is, the more desirable its corresponding decision is. A natural and general functional form of loss is given for speech pattern X of C_k as

$$\ell_k(X, \Lambda) = l(d_k(X; \Lambda)) = \frac{1}{1 + e^{-(\alpha d_k(X;\Lambda)+\beta)}} \quad (\alpha > 0), \tag{4.5}$$

where $l(\cdot)$ is a scalar function that determines the characteristics of the loss function, and α and β are constants. Furthermore, (4.5) is of the sigmoidal form, which is known as a smoothed logistic function, and it is also a smoothed version of the classification error count

$$\ell_k(X, \Lambda) = \begin{cases} 0 \ (C(X) = C_k), \\ 1 \ (\text{others}). \end{cases} \tag{4.6}$$

4.2.5.4 Loss over Multiple Patterns

To determine the performance of a designed classifier over multiple pattern samples, the individual loss is naturally applied in the following empirical average loss form

to a finite (but usually large) set of training pattern samples $\mathcal{X}$ ($= \{X_1, \cdots, X_N\}$):

$$L_0(\Lambda) = \frac{1}{N} \sum_k \sum_n \ell_k(X_n; \Lambda) 1(X_n \in C_k), \tag{4.7}$$

where N is the number of training samples in $\mathcal{X}$, and n of X_n explicitly means that the sample X_n is the n-th sample of $\mathcal{X}$. Assuming the individual loss to be (4.5), this empirical average loss becomes a smoothed version of the total count of classification errors measured over $\mathcal{X}$.

4.2.5.5 Adjustment of Trainable System Parameters

Using the empirical average loss, the design proceeds to an actual training procedure, or, in other words, the adjustment of trainable system parameters. A goal of the adjustment is obviously to achieve the status of classifier parameter, Λ in our case, that results in the minimization of the empirical average loss.

Usually, the functional form of the empirical average loss is unknown, and it is thus rarely possible to achieve the true (global) minimum status of the loss analytically. Accordingly, in most cases, the adjustment is formulated as an asymptotical training method that basically guarantees finding at most local minima of the loss or the global minimum of the loss only in the probabilistic sense. Typical examples of formulated methods include those based on the steepest descent method, the probabilistic descent method, simulated annealing, and genetic algorithms. Among these, we introduce an adjustment method based on the probabilistic descent method [1], which gives a general mathematical framework of adaptive (sequential, or sample-by-sample) parameter adjustment.

A key concept of the probabilistic descent method is that given the loss surface (function), the repetition of small-step descent operations leads at least to a local minimum point of the surface in the probabilistic sense; here, the entire shape of the surface is unobservable. An adjustment rule based on this concept is summarized in the following theorem.

[Probabilistic Descent Theorem]

Assume that a training sample $X(t)$ ($\in C_k$) is given at training time index t. If the classifier parameter adjustment $\delta\Lambda(X(t), C_k, \Lambda(t))$ is specified by

$$\delta\Lambda(X(t), C_k, \Lambda(t)) = -\epsilon(t)\mathbf{U}\nabla\ell_k(X(t); \Lambda(t)), \tag{4.8}$$

and a sequence of positive real numbers in (4.8), $\epsilon(t)$ (referred to as learning weights), satisfies

$$\sum_{t=1}^{\infty} \epsilon(t) \to \infty \qquad \text{and} \qquad \sum_{t=1}^{\infty} \epsilon(t)^2 < \infty, \tag{4.9}$$

then the parameter adjustment according to

$$\Lambda(t+1) = \Lambda(t) + \delta\Lambda(X(t), C_k, \Lambda(t)) \tag{4.10}$$

converges with probability one (1) at least to Λ^*, which results in a local minimum of $L(\Lambda)$, which is the expected loss defined as follows:

$$L(\Lambda) = \sum_k \int_\Omega p(X, C_k)\ell_k(X; \Lambda)1(X \in C_k)dX, \tag{4.11}$$

where $\Lambda(t)$ represents the state of Λ at t, $\mathbf{U}$ is a positive-definite matrix, and Ω is the entire sample space of the X patterns. It is assumed that $dp(X, C_k) = p(X, C_k)dX$, and that $1(\cdot)$ is an indicator function. □

The important points here are that: 1) the adjustment always attempts to remove the classification error caused by a newly presented training sample, and 2) assuming the individual loss to be (4.5), the adjustment can result in the (local) minimization of the expected classification error count loss in the probabilistic sense. The theorem does not guarantee the unconditional minimization of the expected loss, i.e., the achievement of the minimum classification error probability condition. However, it clearly turns out that the discriminative training described in the above paragraphs more clearly resembles the Bayes decision rule (4.1) than the probability function estimation approaches, such as the maximum likelihood method.

4.2.5.6 Training Optimality

The ultimate goal of classifier design is to find the classifier parameter set that achieves the minimum classification error probability condition. However, as cited above, patterns available for training are usually finite, and thus it is difficult to directly aim at the ultimate goal, which would inevitably require the complete computation of the probability related to the sample distribution or its corresponding observation of all possible pattern samples. Actually, training using a limited number of samples basically leads to at most a minimum classification error condition over the finite pattern samples, and this describes little (in a mathematically rigorous sense) about the classification performance over unknown future patterns. Therefore, in order to bridge the ultimate goal, i.e., the minimum classification error probability condition, and the practical design attempts, analyses of the training optimality over infinite training samples are needed to some extent, even though they are only theoretical.

For explanation purposes, let us assume that: 1) a probability measure $p(X)$ is provided in a known functional form for pattern sample X, and 2) a parameter set determining the functional form is $\dot{\Lambda}$. Then, considering the discriminant function

$$g_j(X; \dot{\Lambda}) = p_{\dot{\Lambda}}(C_j \mid X) \tag{4.12}$$

and the misclassification measure of (4.4), we can rewrite the expected loss that is

defined by using the smooth classification error count loss (4.5) as

$$
\begin{aligned}
L(\dot{\Lambda}) &= \sum_k \int_\Omega p(X, C_k) \ell_k(X; \dot{\Lambda}) 1(X \in C_k) dX \\
&\simeq \sum_k \int_\Omega p(X, C_k) 1(X \in C_k) \\
&\quad \times 1 \left(p_{\dot{\Lambda}}(C_k \mid X) \neq \max_j p_{\dot{\Lambda}}(C_j \mid X) \right) dX. \qquad (4.13)
\end{aligned}
$$

Obviously, the last expression is equivalent to the expected error caused by the classification decision using the estimates of the a posteriori probabilities. Note here that the parametric form $\dot{\Lambda}$ is known, but it is not known whether the present status of $\dot{\Lambda}$ results in the minimum error-rate classification. Here, the difference in the near equality of (4.13) originates in the smoothness contained in the error count loss. Consequently, by controlling the smoothness of functions such as the L_p norm used in (4.4) and the sigmoidal function used in (4.5), we can arbitrarily make $L(\dot{\Lambda})$ closer to the last equation in (4.13). It should be noted here that there is a link between the practical design operation based on the discriminative training and the computation of the expected loss, which is the ultimate theoretical goal of classifier design.

Next, let us recall that we use $\dot{\Lambda}$, whose functional form is assumed to be known. Based on this fact and also the relation between the minimum error-rate classification and its corresponding a posteriori probabilities, the status of $\dot{\Lambda}$ that corresponds to the minimum of $L(\dot{\Lambda})$ in (4.13) (which is achieved by adjusting $\dot{\Lambda}$) is clearly equal to the $\dot{\Lambda}^*$ that achieves the maximum a posteriori probability condition. In short, it turns out that the minimum condition of $L(\dot{\Lambda})$ can become arbitrarily close to the ideal, minimum error-rate condition that is associated by the minimum classification error probability

$$
\mathcal{E} = \sum_k \int_{\Omega_k} p_{\dot{\Lambda}^*}(X, C_k) 1(X \in C_k) dX, \qquad (4.14)
$$

where Ω_k is a partial space of Ω that causes a classification error according to the maximum a posteriori probability decision rule, i.e.,

$$
\Omega_k = \left\{ X \in \Omega \,|\, p_{\dot{\Lambda}^*}(C_k \mid X) \neq \max_j p_{\dot{\Lambda}^*}(C_j \mid X) \right\}. \qquad (4.15)
$$

The above analysis of the minimum error condition is important from the theoretical viewpoint of showing the rationality of the discriminative training formalism. It provides a mathematically sound background to practical attempts at classifier design based on discriminative training. In reality, however, the parameter set $\dot{\Lambda}$ is rarely known, and thus it is usually impossible to achieve the minimum classification error-rate condition through discriminative training, or, in other words, the loss minimization over finite training samples.

4.2.5.7 Global Design Scope

We have assumed that a feature extraction module is determined separately from the design of a classifier, but this assumption is only adopted to simplify discussion. Actually, feature extractors are usually designed based on scientific expertise, and such separate and empirical design is a standard approach to feature extractor design (as a result, recognizer design). However, from the formalism introduced above, one can easily conclude that the design scope of the discriminative training for a back-end classifier can be extended to its corresponding front-end feature extractor. A key in this extension is clearly to use the chain rule of calculus. Recall that this rule plays a central role in the definitions of the misclassification measure and the individual loss.

From the definition of our recognizer structure, a discriminant function for the entire recognizer can be defined as

$$g_j(U; \Psi) = g_j(f(U; \Phi); \Lambda), \tag{4.16}$$

where $f(\cdot)$ is an output of the feature extraction module. Clearly, the adjustment that is performed for the loss minimization at the level of $g_j(\cdot; \Lambda)$ is easily propagated to the level of $f(U; \Phi)$, leading to a global optimization of the entire system.

4.3 Speech Recognizers Based on Neural Networks

4.3.1 Preparations

In the previous section, we summarized the Bayes decision theory, in particular the discriminative training that underlies the NN-based speech pattern recognition. From those descriptions, one can see that there can be various kinds of embodiments and implementations of the training. For example, by controlling μ in (4.4), one can achieve various implementations of the misclassification measure, and by controlling α and β in (4.5), one can also achieve various shapes of the smoothed classification error count loss. In addition, there are obviously many other possibilities of defining individual loss in place of the sigmoidal form loss of (4.5). Actually, in the history of NN-based speech pattern recognition, loss functions other than the smoothed error count loss have been rather popular. The most typical selection from such popular functions is the squared error loss, which is defined between a classifier output and its counterpart target (teaching) signal. The second typical selection is the cross entropy loss, defined by using the estimates of the class-conditional probabilities. To introduce NN-based speech pattern recognizers, we thus need to refer to the definitions of these popular loss functions.

As previously stated, a speech signal is dynamic. On the other hand, a basic structure of NN, such as a standard multi-layer Perceptron network, is set for handling static (fixed-dimensional) vector patterns. Indeed, one of the important aspects of NN-

based speech recognition research has been to cope with the discrepancy between the dynamic nature of speech signals and the NN's traditional structure, which is only suited for static patterns. In the early stage of research, shift-tolerant structures were examined for classifying short segments of speech signals. Next, networks having both a standard structure suited for static patterns and a shift-tolerant structure were used in a hybrid form with the standard speech pattern classifier structure of HMM. In addition, network structure itself was further examined, resulting in the development of the recurrent network, which possesses recurrent signal flows in order to represent the temporal structure of speech.

Obviously, there are many possibilities in combining the loss functions and the NN structures. Actually, many types of combinations have been reported, and it is difficult to introduce them comprehensively in the limited space of this chapter. Therefore, we selectively focus on several historic cases of NN-based speech recognition attempts. Introductions will be organized along the line of the loss and NN structure selections.

4.3.2 Classification Error Minimization

4.3.2.1 Learning Vector Quantization

Learning vector quantization (LVQ) is one of the pioneer NN-based pattern classifiers [18, 19]. Originally, it was developed in the framework of a self-organizing feature map that simulated the physiological representation of memory. However, its behavioral principle is simple and can be considered an adaptive training of reference vectors, each of which basically represents a class model in a preset distance space.

Several versions of LVQ have been proposed. In the speech recognition field, the second version of LVQ, i.e., LVQ2, has been the most extensively applied [18]. In addition, the differences among the various LVQ's are insignificant from the viewpoint of discriminative training formalism. Therefore, here we summarize the training principle of LVQ2 and discuss it from the loss selection viewpoint of the discriminative training.

A classifier to be trained with LVQ2 assumes an input to be a static vector and also assumes each class to be modeled by multiple reference vectors, each being in the same vector space as the input. In the classification stage, an unknown input vector is classified as the class of the reference vector that has the smallest distance to that input vector. This classification scheme means partitioning the vector space into regions defined by individual reference vectors, or, in other words, vector quantization of the original vector space. LVQ2 training adjusts the reference vectors so that each input vector has a reference vector of the right class as its closest reference vector.

More precisely, LVQ2 training is summarized as follows. For a given training input vector $\mathbf{x}$ of C_k, three conditions must be met for training to occur: (1) the nearest class must be incorrect, (2) the next-nearest class must be correct, and (3) the training vector must fall inside a small, symmetric window defined around the mid-plane of the reference vectors $\mathbf{r}_j$ ($\in C_j$ being an incorrect class) and $\mathbf{r}_k$ ($\in C_k$ being the correct class). If these conditions are met, the incorrect reference vector is moved

further away from the input, while the correct reference vector is moved closer, according to

$$\begin{cases} \mathbf{r}_j(t+1) = \mathbf{r}_j(t) - \alpha(t)(\mathbf{x}(t) - \mathbf{r}_j(t)), \\ \mathbf{r}_k(t+1) = \mathbf{r}_k(t) + \alpha(t)(\mathbf{x}(t) - \mathbf{r}_k(t)), \end{cases} \tag{4.17}$$

where (t) means the status of its corresponding vector at time index t, and $\alpha(t)$ is a monotonically decreasing, small value function of the time index.

One may notice here that the adjustment rule of LVQ2 is similar to the training rule defined by the probabilistic descent theorem. Actually, it was demonstrated that LVQ2 was a heuristic and modified version of GPD training using the smoothed classification error count loss, i.e., (4.5), while letting μ go to ∞ with some modification in the definition of the misclassification measure, for a multiple reference distance classifier [12]. See [12] and [17] for details. In short, LVQ2 is an NN-motivated discriminative training method that aims at the minimization of the (smoothed) classification error count loss.

4.3.2.2 Shift-tolerant LVQ Classifier

In the simplest case, an LVQ-trained distance classifier is applied to a static feature vector, which is calculated for every time window position over a dynamic input speech pattern. For every feature vector, LVQ2 training is executed, and the classification using the trained reference vectors is performed. However, individual feature vectors are often insufficient for achieving correct speech pattern classification at a meaningful speech unit level such as phoneme or word. To cope with this insufficiency, LVQ2 with a shift-tolerant architecture was first applied to phoneme pattern classification [20].

Figure 4.1 illustrates the architecture of a shift-tolerant LVQ (STLVQ) system for classifying three phoneme classes. It is assumed here that an input speech pattern is a priori segmented and labeled with its corresponding correct phoneme class. In the classifier, each class is assigned a number of reference vectors. The LVQ2 training procedure is applied to speech feature vector patterns that are stepped through in time. For every feature vector position, the classification decision is evaluated by using the phoneme label, and reference vectors are adjusted according to the LVQ2 training rule. In the classification stage for an unknown input, a slightly different procedure than simply finding the closest vector to the input is employed: The shift-tolerant architecture produces several closest reference vectors, one for each window position. The procedure can be summarized as follows: (1) for each window position and for each class, the classifier calculates the distance between the input vector and the closest reference vector within one class, (2) from this distance measure, each class is assigned an activation value that is high for small distances, low for large distances, (3) after the window has been shifted over the entire input pattern, the activations calculated at each window position are summed for each class, (4) the class with the highest overall activation is chosen as the encoded class.

Experimental evaluations of the STLVQ classifier are reported in detail in [20].

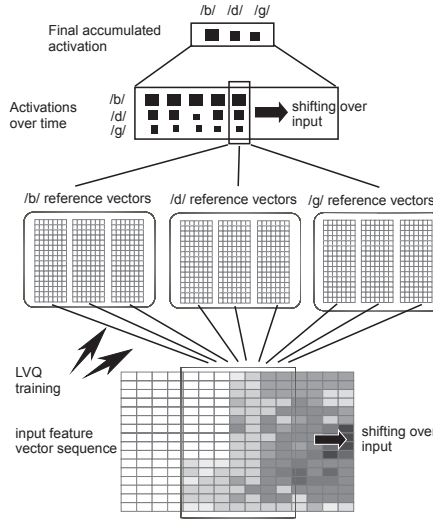

FIGURE 4.1
Architecture of shift-tolerant LVQ classifier [20].

4.3.2.3 LVQ/HMM Hybrid Classifier

The shift-tolerant architecture is useful for alleviating the limitation of the original LVQ-trained pattern classifier, which is suited only for static vector patterns. However, the usual goal of speech pattern recognition is to encode a natural-length speech input to its corresponding word or sentence (word sequence) class. Obviously, the shift-tolerant architecture is insufficient for achieving this type of goal.

A straightforward solution is to combine LVQ and HMM. Traditionally, HMM has been trained in the probability function estimation approach, where the classification power is usually poorer than the discriminative training. Thus, a hybrid of LVQ and HMM is a natural and promising choice: LVQ is an embodiment of discriminative training.

There are two types of HMM: (1) discrete HMM having a codebook for dealing with an input as an observation of the multinomial distribution, and (2) continuous HMM dealing with an input as an observation of a continuous probability function such as the Gaussian probability function. Because the codebook is equivalent to a pool of reference vectors, the LVQ/HMM hybrid naturally uses the discrete HMM for its implementation [9].

Figure 4.2 illustrates an LVQ/HMM hybrid speech pattern classifier. The system consists of a codebook and a state transition Markov chain. The codebook contains a number of pairs of codes (symbols) and code vectors. The code vectors are set in the same vector space as an input feature vector, and they are used to encode the

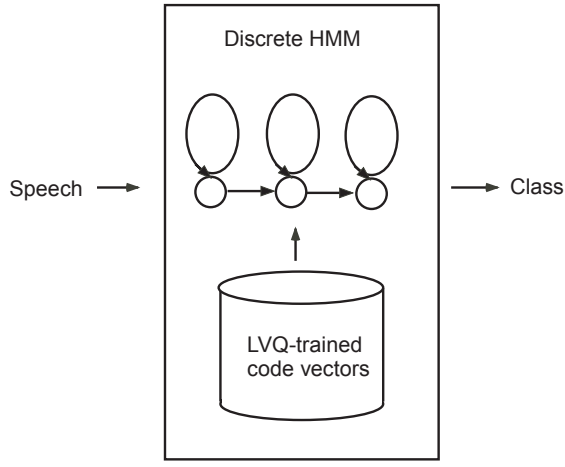

FIGURE 4.2
Block diagram of LVQ/HMM hybrid classifier.

input to the code having the closest code vector to the input. The Markov chain module then treats the selected code as an observation of the discrete multinomial probability model. LVQ algorithms are applied to the design of the code vectors, while the code vectors are conventionally determined by using clustering methods, such as the k-means method.

A standard design objective of conventional clustering methods is to minimize the average distortion between the code vectors and training input vectors. This design does not necessarily increase the discriminative power of the codebook. If code vectors contain discriminative information that is useful for the classification of phoneme units or sub-phonemic units, the post-end HMM module can fundamentally make more accurate classification decisions for the entire input utterance (word or word sequence) pattern. In [9], LVQ/HMM was implemented by using LVQ2, and its high discriminative power was successfully demonstrated.

4.3.2.4 HMM/LVQ Hybrid Classifier

Discriminative power should be incorporated in a stage that is as close as possible to the final classification decision of a recognizer. Obviously, a phoneme classifier should have high discriminative power for phoneme classification, a word classifier should have high discriminative power for word classification, and a word sequence classifier should have high discriminative power for word sequence classification. In light of this, the LVQ/HMM classifier is insufficient for making the best use of the

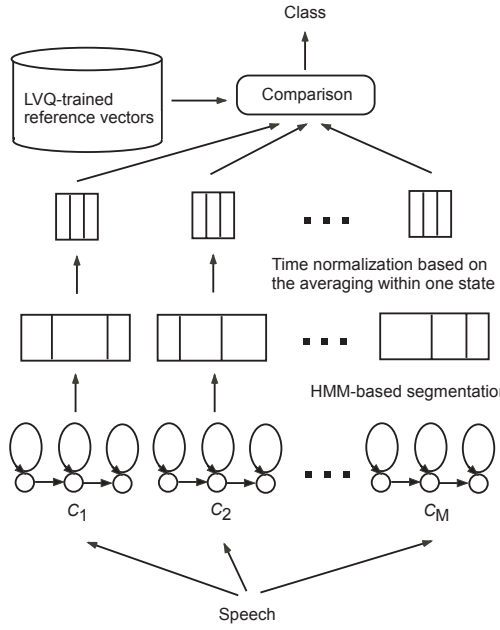

FIGURE 4.3
Block diagram of HMM/LVQ hybrid classifier.

LVQ's discriminative capability. This concern suggests a reversal of the hybrid idea, i.e., HMM/LVQ [13].

Figure 4.3 illustrates an HMM/LVQ hybrid classifier. HMM is used here to normalize the nonlinear temporal structure of speech inputs, and an LVQ-trained distance classifier works as a system of classifying a duration-normalized input pattern. The system includes a set of n-state HMMs (one HMM per speech unit to be classified, e.g., word class) in addition to the LVQ classifier. Reference vectors of the LVQ classifier are set in the vector space of the duration-normalized input patterns.

The training procedure of the system is divided into the following two subsequent procedures: (1) temporal normalization, and (2) LVQ training of reference vectors. The mechanism of temporal normalization is as follows: (1) HMMs are trained in a regular manner (usually based on the MLE method) for their corresponding classes, (2) each HMM makes state-based segmentation by using the Viterbi segmentation, (3) the average vector is calculated over the feature vectors assigned to one state, so that in the case of n-state HMM, an input (F-dimensional) vector sequence is mapped to an $(F \times n)$-dimensional vector (precisely, $F \times n$ matrix), and (4) assume that M class HMMs are available. A time-normalized vector is assigned a class

label of the HMM used for its generation. Then, M time-normalized vectors are generated, each treated as a different class token. LVQ training, especially LVQ2 in [13], designs reference vectors so that the classifier can correctly classify all of the generated tokens.
In the classification stage for an unknown input, the input is first converted to M tokens, each labeled as a different class. Then, for every token, the nearest reference vector is selected, and the distance between the nearest vector and the token is calculated. Because M tokens are generated here, M distance values are calculated. Finally, a classification decision is made for the input speech pattern by using a weighted sum of the distance values.
The high discriminative power of this hybrid classifier was shown in a confusable American E-rhyme phoneme recognition task [13].

4.3.3 Squared Error Minimization

4.3.3.1 Training Using the Squared Error Loss

For the exemplar M class task, the squared error loss is defined as

$$\ell_k(X;\Lambda) = \frac{1}{2}\sum_{j=1}^{M}\{g_j(X;\Lambda) - \tau_j\}^2, \tag{4.18}$$

where $\{\tau_j\}$ is a teaching (target) signal that is usually set, for a training sample X ($\in C_k$), to

$$\tau_j = \begin{cases} 1 & (j = k), \\ 0 & (\text{otherwise}). \end{cases} \tag{4.19}$$

The loss is then rewritten as

$$\begin{aligned} \ell_k(X;\Lambda) &= -g_k(X;\Lambda) + \frac{1}{2}\{g_k(X;\Lambda)\}^2 \\ &\quad + \frac{1}{2} + \frac{1}{2}\sum_{j,j\neq k}^{M}\{g_j(X;\Lambda)\}^2 \\ &> -g_k(X;\Lambda) + \frac{1}{2}\sum_{j,j\neq k}^{M}\{g_j(X;\Lambda)\}^2. \end{aligned} \tag{4.20}$$

Here, one can find that the bottom line expression of (4.20) can be treated as a kind of misclassification measure:

$$d_k(X;\Lambda) = -g_k(X;\Lambda) + \frac{1}{2}\sum_{j,j\neq k}^{M}\{g_j(X;\Lambda)\}^2. \tag{4.21}$$

It turns out that the reduction of the squared error loss results in the reduction of this misclassification measure, and also that the minimization of the squared error loss of (4.18) is equivalent to the minimization of a simple linear loss in which the

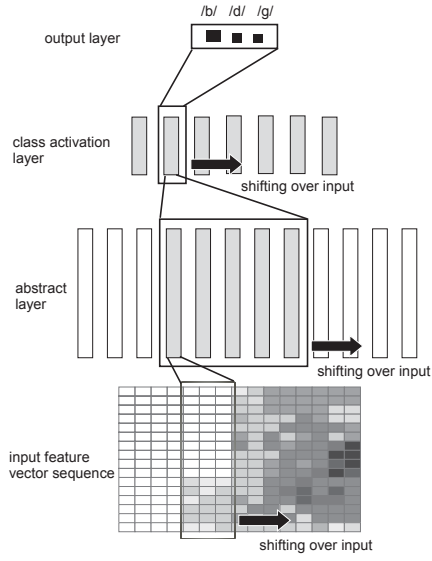

FIGURE 4.4
Architecture of time-delay neural network [27].

misclassification measure (4.21) is embedded. Thus, the training using the loss (4.18) is certainly discriminative but obviously different from the attempt to achieve the minimum classification error condition by using the smoothed error count loss. See [17] for detailed discussions.

4.3.3.2 Time-delay Neural Network

The time-delay neural network (TDNN) is one of the classic NN applications to speech pattern recognition [27]. The time-delay architecture is incorporated to cope with the speech signal dynamics by using multiple feature vectors, each being generated by shifting a time window over an input speech pattern. That is, this design is a shift-tolerant architecture, and it was a model for the development of the shift-tolerant LVQ classifier [20].

Figure 4.4 illustrates a typical example of the TDNN architecture. The TDNN classifier assumes an input speech pattern to be a sequence of feature vectors. It uses a limited-length stream of the feature vectors as its input, and in order to feed forward the information of a short, focused segment, it constraints network connections to a smaller number of nodes than the number of frames in the whole limited stream (for explanation purposes, let us call this set of limited connections a delay-group.). The classifier then accumulates the information fed forward from the lower network layers by shifting the delay-group over the input.

Training of the TDNN classifier is performed with the squared error loss. A training

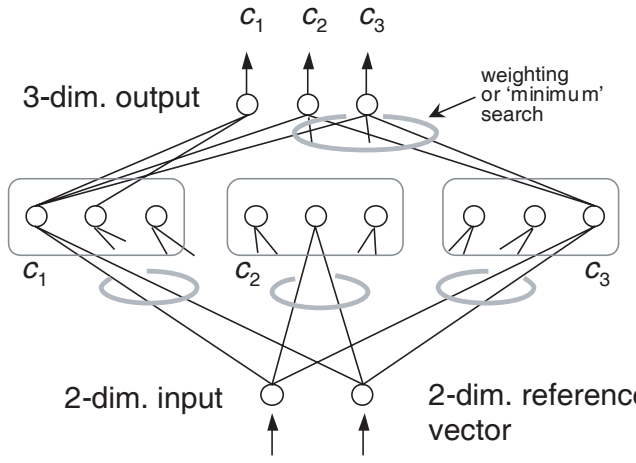

FIGURE 4.5
Schematic description of distance classifier as a single intermediate layer network (2-dimensional input, 3 references/class, 3 classes).

target is a vector, of which the component for the correct class is one (1) and of which components for other classes are set to zero (0). Then, the loss is defined between the target and an output of the classifier. Accordingly, the minimization of this squared error loss works so that the classifier output resembles the training target. In the classification stage for unknown patterns, the class having the largest network output is selected as a classification result.

As shown in Figure 4.4, the TDNN classifier has multiple intermediate layers. This multi-intermediate-layer structure is a special feature of multi-layer networks, such as MLP, and it is clearly distinct from the structure of traditional classifiers, such as a distance classifier only using reference vectors as class models, which can be considered a single intermediate layer network (Figure 4.5). In addition, the reader could recall the case of an LVQ classifier that has only reference vectors. A principal function of the multiple intermediate layers is information abstraction. In the TDNN classifier, it is expected that information essential for classification, which is accordingly shift-tolerant, can be extracted through these hidden layers.

In [27], highly influential experimental results of using the TDNN classifier for phoneme classification are reported.

4.3.3.3 Multi-state Time-delay Neural Network

To cope with the dynamic nature of longer speech units, the concept of TDNN was directly extended to a mutli-state time-delay neural network (MSTDNN) [8]. This

development contrasts with that of the hybrid systems using LVQ and HMM. It can be said that MSTDNN incorporates the state transition structure, which was widely used in graphical models such as HMM, in its system structure in place of simply combining existing concepts such as TDNN and HMM.

An MSTDNN classifier consists of a number of local TDNN systems, each of which forms a state and is designed for the classification of phonemes or sub-phonemic units, and it outputs a classification result for the entire speech input, e.g., word sequence classification result through the DP-based search of the best phoneme state sequence.

Training of the classifier is done with the squared error loss, as for the original TDNN classifiers. However, the loss here is defined at the level of the final class outputs of the task, i.e., word or word sequence classes, in place of the short phoneme level, and additionally the DP-based segmentation over the entire training input pattern is embedded in the loss minimization. Clearly, the training of MSTDNN is suitable for the classification of longer speech units. Indeed, as an alternative to the standard HMM-based classifier, the MSTDNN classifier has been used in various speech pattern recognition tasks [7].

4.3.4 Cross Entropy Minimization

4.3.4.1 Training Using the Cross Entropy Loss

There is another interpretation of the target signal (4.19), which was introduced for training by minimizing the squared error loss. That is, the j-th component of the target signal can be considered the a posteriori probability of class j, which is represented as the multinomial distribution, where the probability of X belonging to a correct class is one (1) and those of X belonging to other classes are zero (0). Then, to make discriminant function for X resemble this a posteriori probability, the following cross entropy loss has been employed with the assumption of using a softmax network output function:

$$\begin{aligned} E &= -\sum_{n=1}^{N}\sum_{j=1}^{M} \tau_j \ln\{g_j(X_n;\Lambda)\} \\ &= -\sum_{n=1}^{N} \ln \frac{\zeta_{\kappa(n)}(X_n;\Lambda)}{\sum_{m=1}^{M}\zeta_m(X_n;\Lambda)} \\ &= -\sum_{n=1}^{N}\left[\ln\left\{\zeta_{\kappa(n)}(X_n;\Lambda)\right\} - \ln\left\{\sum_{m=1}^{M}\zeta_m(X_n;\Lambda)\right\}\right], \end{aligned} \tag{4.22}$$

where X_n is assumed to belong to $C_{\kappa(n)}$, and $\zeta_{\kappa(n)}(X_n;\Lambda)$ is an input to the $\kappa(n)$-th output node, which corresponds to $C_{\kappa(n)}$ and $g_{\kappa(n)}(X_n;\Lambda)$, of the classifier. It is assumed that this node produces a softmax value, $\zeta_{\kappa(n)}(X_n;\Lambda)/\{\sum_{m=1}^{M}\zeta_m(X_n;\Lambda)\}$. One may note here that the subtraction form in the bottom line of (4.22) can be

considered to be a type of misclassification measure as follows:

$$d_{\kappa(n)}(X_n;\Lambda) = -\ln\left\{\zeta_{\kappa(n)}(X_n;\Lambda)\right\} + \ln\left\{\sum_{m=1}^{M}\zeta_m(X_n;\Lambda)\right\}. \tag{4.23}$$

(4.23) represents an operation of discriminant function comparison over the possible classes, and therefore it can certainly be considered a discriminant function. Then, the average loss E to be minimized becomes

$$E = \sum_{n=1}^{N} d_{\kappa(n)}(X_n;\Lambda). \tag{4.24}$$

Observation here tells one that this loss can also be treated as a case of applying the linear individual loss function to the misclassification measure (4.23). Due to the employment of the linear loss, there seems to be a discrepancy between the minimization of the cross entropy loss and the achievement of the minimum classification error probability condition.

4.3.4.2 Unidirectional Network Classifier

The cross entropy loss was used in an early study that pioneered the application of recurrent neural networks to speech pattern recognition [23].

Natural signals such as speech signals obey the law of causality. Thus, in principle, temporal information goes from the past to the future, and accordingly a unidirectional (past-to-future, or, left-to-right) network structure is often used as a primary selection for modeling such signals. A typical structure of a unidirectional network is illustrated in Figure 4.6. In this figure, at time index t, input acoustic vector $\mathbf{u}_t$ is presented to the network along with the state vector $\mathbf{s}_t$, and these two vectors produce the output vector $\mathbf{y}_t$ and the next state vector $\mathbf{s}_{(t+1)}$. A design goal here is to determine two weight matrices, $\mathbf{W}$ and $\mathbf{V}$, so that $\mathbf{y}_t$ can satisfy a preset design objective.

In [23], the cross entropy loss was applied to a recurrent network used as a likelihood estimator in conjunction with an HMM-based speech pattern classifier. The network training for minimizing this loss was done with a standard training method for recurrent networks, i.e., the back-propagation through time method that expands a recurrent network in time or, in other words, considers a recurrent network for all time indices as a single very large network with an input and output at each time index and shared weights over all time indices.

4.3.4.3 Bidirectional Network Classifier

In principle, the temporal correlation based on causality is represented in the forward unidirectional information flow. In addition to this correlation, speech signals usually possess backward directional temporal correlation: The speech signal is an output of a physiological articulation system that is controlled by a speech production plan, which prepares future articulation and, accordingly, has backward influence on the

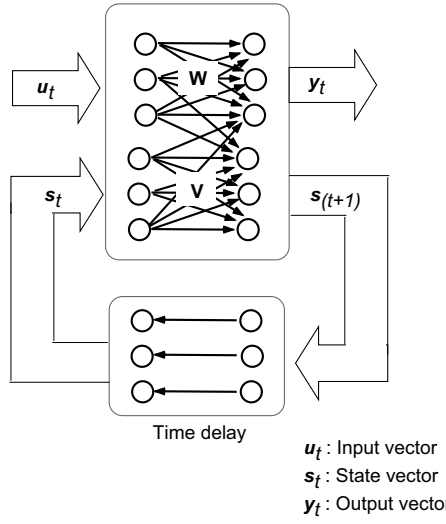

FIGURE 4.6
Architecture of unidirectional network [23].

past acoustic state of the speech signal. To represent these two information flows, a bidirectional network was introduced, and it was designed with the minimization of the cross entropy loss [25].

A sample structure of a bidirectional network is illustrated in Figure 4.7. A key idea in the structure is to split the state neurons into two parts: one part responsible for the forward time direction (forward states) and the other part responsible for the backward time direction (backward states). It should be noted here that there is no interaction between the two differently directional networks and therefore each can be designed in the same way as a unidirectional network, e.g., by using the back-propagation through time method. In [25], the utility of the bidirectional network was demonstrated in phoneme pattern classification.

4.4 Fusion of Multiple Classification Decisions

4.4.1 Principles

Using many decisions is generally more stable and often more useful, in terms of robustness to unknown pattern samples that do not appear in the training/design stage,

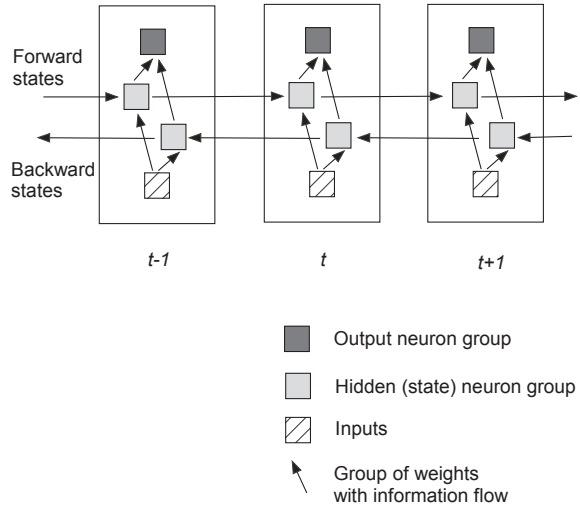

FIGURE 4.7
Architecture of bi-directional network [25].

than using a single decision. In real-world pattern classification problems, true sample distributions are generally unobservable, and achievable classification decisions are merely estimates of the true classification decisions, each relying on its corresponding true class boundaries. Therefore, the proverb basically holds true even in the scientific framework of pattern classification, and the concept of decision fusion, i.e., making a final decision by combining multiple pre-decisions, has recently attracted many researchers' interests.

The simplest and most basic way of decision fusion is to use the average of multiple pre-decisions. Intuition suggests that the decision made by averaging independent pre-decisions is more stable (insensitive to the selection of training samples or robust to unknown testing samples) than the individual pre-decisions included therein. Actually, based on the tractable nature of the square error loss, the decision fusion methods using this loss function have been extensively explored in the literature (e.g., [26]), and it has been shown that in principle the decision fusion scheme reduces errors in regression estimation. Assuming the target to estimate to be the multinomial distribution function that represents the class index information of samples, the analysis results of the decision fusion mechanism for regression cases can be applied to the cases of classification. However, it should be recalled that there is a discrepancy between the training with the minimization of the squared error loss and the achievement of the minimum classification error probability condition. Thus it seems that

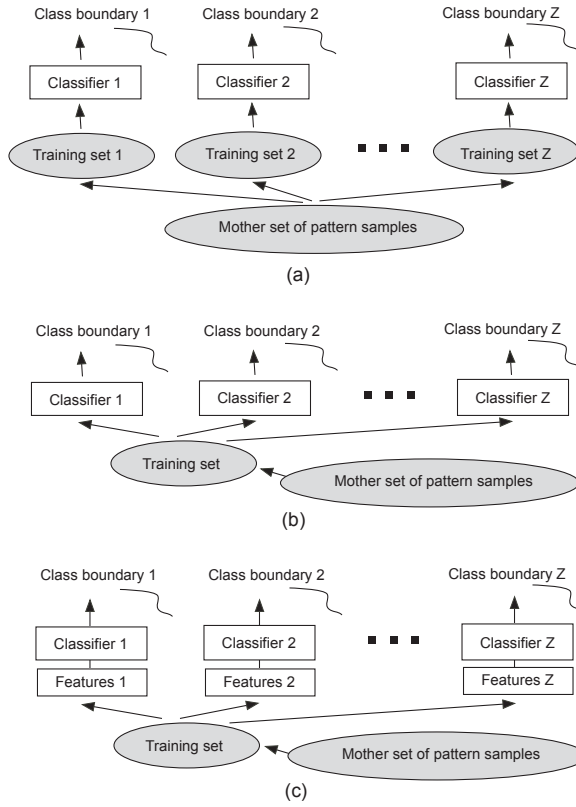

FIGURE 4.8
Typical classifier design schemes of averaging-based decision fusion.

the simple application of the results of regression problems to classification problems is insufficient, and that it is necessary to further analyze decision fusion formalisms that can be more directly applied to the minimization of classification errors.

As one might imagine from the discussions in the above paragraph, the decision fusion approach to speech pattern recognition has been tested in somewhat heuristic styles. Most cases of the approach have empirically employed the averaging or weighted averaging scheme of pre-decisions, simply expecting that the resulting decisions would be more robust to unknown pattern samples. Figure 4.8 illustrates three major types of embodiments of the averaging-based decision fusion for classification: (a) design of multiple sub-classifiers using different sets of training data, (b) design of multiple sub-classifiers using a single training data set, and (c) design of multiple sub-classifiers using different types of feature representation. In all three cases, a final decision is made through averaging pre-decisions, each made with its

corresponding sub-classifier. In other words, the final estimated class boundary is determined by averaging the pre-boundaries, each produced by its corresponding sub-classifier. The pattern sample sets for training are the ones extracted from the mother sample set, which is usually unobservable. In (a), all of the training sample sets use a common method of feature representation, that is, all of the patterns of the training sample sets are represented in an identical feature space. Here, each training sample set is extracted independently from the mother data set, and a sub-classifier is designed by using only its corresponding training data set. In contrast, in (b), multiple sub-classifiers are designed over a single training data set (of some finite size), aiming to maintain the independency among the design procedures of the different sub-classifiers. In (c), a different sub-classifier is designed by using a different feature representation. Here, the different types of features are calculated over a single training sample set (of some finite size). Because the effect of the averaging scheme assumes that the sub-classifiers are designed independently, design procedures, which are illustrated by arrows in the figure, should be basically as independent as possible. In this light, case (a) seems to be the most effective for utilizing the value of the decision fusion scheme. However, preparing multiple independent training sample sets is often costly, and most cases of speech pattern recognition have employed cases (b) and (c). In the remaining pages of this section, we shall introduce several exemplar embodiments of the averaging-based decision fusion approach to speech pattern recognition.

4.4.2 Examples of Embodiment

4.4.2.1 Multi-codebook Classifier Designed with GPD

One of the most straightforward embodiments of case (b) in Figure 4.8 was developed in [3]. In this report, the GDP training was applied to prototype- (reference vector)- based distance classifiers, which are basically the same as an LVQ classifier. In an experimental task of classifying 9-class American E-rhyme speech samples, it was shown that the use of multiple codebooks, each designed separately with GPD, could successfully increase the classification accuracy of the baseline single GPD-trained codebook classifier; the accuracy of the baseline system, i.e., 73%, was increased to 80%, which was among the highest scores on the E-rhyme data set. Here, each of the multiple codebooks is of the same size as the codebook of the baseline single codebook system.

In [3], another effect of the decision fusion was demonstrated. Figure 4.9 shows classification accuracy scores as a function of the number of prototypes per class and codebook. Here, the total number of prototypes per class was kept constant, but the number assigned per codebook was changed. For example, with the total of 24 prototypes per class, one could have 4 codebooks with 6 prototypes per class, or 8 codebooks with 3 per class. The curve in the figure clearly shows that there is a balance between the fine-grained boundary estimation of many prototypes per codebook and the coarse-grained averaging with many codebooks of fewer prototypes, suggesting the utility of averaging scheme under a practical condition where

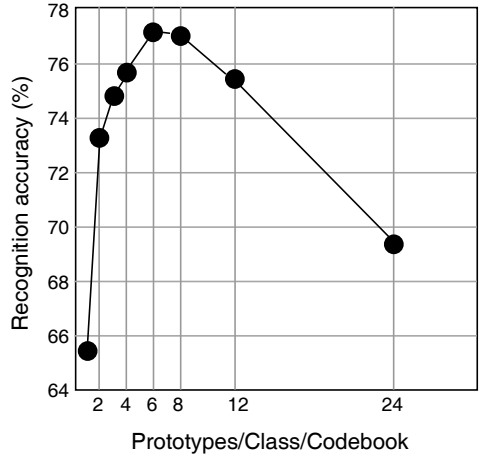

FIGURE 4.9
Relation between recognition accuracy and the number of prototypes per class and codebook [3].

available system resources such as prototypes are limited.

4.4.2.2 Multi-class Classification Based on Support Vector Machine

Generally, classification in a high-dimensional vector space is easier than in a low-dimensional vector space. By focusing on this general nature of classification, the support vector machine (SVM) has been attracting much research interest recently. As shown in the following paragraphs, SVM-based multi-class classification can be considered a type of embodiment of case (b).

The purpose of SVM-based classification is to convert a given task of classifying samples in some originally low-dimensional vector space to a task of classifying the samples in a high-dimensional vector space through a nonlinear sample projection performed by a neural network's kernel function. The projected samples are then encoded as one of two classes by using a linear discriminant function having the largest margin in a two-class boundary region. Thus, this approach is naturally expected to possess two types of robust properties: one based on the large-margin classification in a high-dimensional vector space and the other based on the fusion of two-class classification decisions for a multi-class task setting. It would seem that the first type of robustness originates as a means to circumvent outlier samples that tend to be misclassified, and the second originates from the increased statistical stability in setting class boundaries.

As its name implies, SVM is a method for classifying fixed-dimensional, static vector patterns. Extension of this methodology to the classification of dynamic patterns is

still an ongoing research topic, and therefore, applications of SVM to speech pattern recognition are not yet that mature. In the following paragraph, we introduce one recent application example of SVM classifiers in [2].

The task tested in [2] was classification of phoneme segments that were represented as static feature vectors whose components were formant (acoustic resonance) frequency values or average cepstral coefficients. SVM was applied to these static vector patterns. Because the SVM formalism originally assumes the number of classes to be two, there are two possible combinatorial formations of classification: (1) the "one vs. one" formation and (2) the "one vs. all" formation. In the one vs. one formation, an SVM-based sub-classifier was designed for every pair of two different classes. An unknown test sample was preclassified by all of the designed sub-classifiers and was then finally classified with a voting scheme over the sub-classifiers. Here, the voting scheme is equivalent to averaging over pre-decisions. In the one vs. all formation, an SVM-based sub-classifier was designed for every pair of a target class and remaining classes. A test sample was preclassified by all of the sub-classifiers and then finally encoded to the class having the largest distance from the separating hyperplane. It should be noted that this second scheme does not include the effect of the averaging. These two types of SVM-based classifiers were compared to a conventional Gaussian mixture classifier, and the superiority of the one vs. one formation using SVM was demonstrated.

The above one vs. one formation incorporated by the voting scheme is a typical framework of decision fusion. However, it is not trivial to show the superiority of the framework theoretically. An important ongoing research issue is how to apply the SVM-based two-class classifier to multi-class speech pattern classification tasks.

4.4.2.3 Decision Fusion Using Different Classifiers

In the embodiments of the previous subsections, a single type of classifier structure was employed. For example, all of the sub-classifiers in Section 4.4.2.1 are prototype-based distance classifiers, and all those in Section 4.4.2.2 are SVM-based classifiers. A different type of classifier could represent a different type of class boundary (accordingly, classification decisions). In light of this, another type of embodiment of case (b) in Figure 4.8, i.e., one combining different types of classifiers such as HMM and neural network, has been tested in an important sub-area of speech pattern recognition, i.e., speaker recognition (e.g., [5]).

In [5], three types of classifiers were used: an HMM-based system, a DP-based distance classifier, and a neural tree network system, which is a hierarchical classifier that uses a tree architecture to implement a sequential linear decision strategy. Through comprehensive experimental evaluations, a basic tendency was demonstrated: the larger the number of sub-classifiers is, the more accurate the final fused classification decisions are. An overview of the decision fusion approach to speaker recognition can be found in the literature (e.g., see [6]).

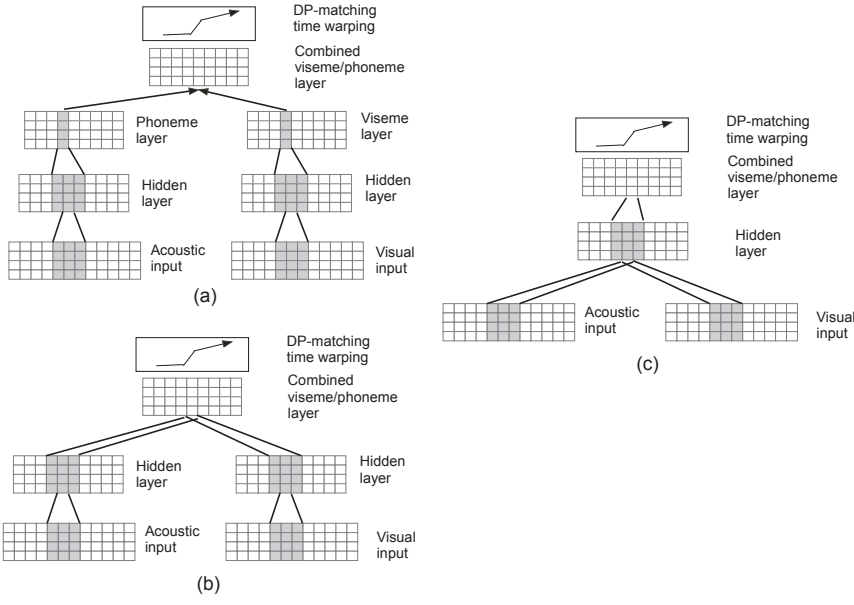

FIGURE 4.10
Typical block diagrams of the MSTDNN-based audio-visual speech recognition [7].

4.4.2.4 Decision Fusion Using Multi-modal Classifiers

The contribution mechanism of the decision fusion, case (c) of Figure 4.8, is basically different from that of the previous two cases, i.e., (a) and (b). In case (c), the decision combination is performed in different feature spaces and thus its resulting effect is not the same as the average computation in a single feature space. In this multi-feature case, the sub-classifiers, each designed over a different feature set, are expected to mutually compensate the weakness of their competing sub-classifiers, while the sub-classifiers in cases (a) and (b) are expected to participate the averaging operation in order to reduce the statistical variance of the individual pre-decisions.

Among the many possibilities of the decision fusion using different types of features, the use of a visual face image as well as its corresponding acoustic speech signal has been rapidly growing as an emerging research topic. Perfect recognition of speech signals is essentially difficult, due to several reasons such as the incompleteness of articulation and acoustic distortion over speech transmission channels. Even for humans, hearing over a telephone is usually more difficult than hearing in face-to-face communication, probably due to the lack of face information. In the following, we introduce two recent examples of audio-visual speech recognition that uses lip shape information in addition to the standard observation of a speech signal [7, 21].

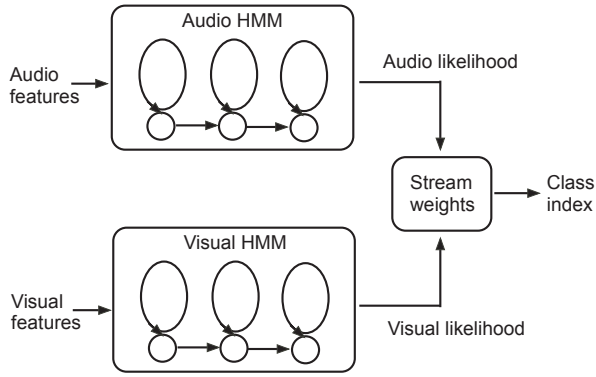

FIGURE 4.11
Block diagram of the twofold-HMM-based audio-visual speech recognition [21].

In [7], MSTDNN classifiers were used to classify an input stream that consisted of an acoustic speech signal and its corresponding visual lip shape image signal. Based on the structural flexibility of MSTDNN, there are several possibilities of signal combination. Three types of combination, illustrated in Figure 4.10, were actually examined: (a) data fusion of combining audio-visual data on the viseme/phoneme layer, (b) data fusion of combining audio-visual data on hidden layer, and (c) data fusion of combining audio-visual data on the input layer. Among these, in principle, the structure of (a) is the best suited for independent design of the two sub-classifiers, each for a different type of modal data. Independency here allows one to carefully train each sub-classifier, reflecting the nature of each modal data in its corresponding design of the sub-classifier. Comprehensive experimental evaluations clearly demonstrated the effect of the data fusion over the multi-modal data stream, specially the approach of using the type (a) network.

In [21], in place of the so-called neural networks, HMM's were used as sub-classifiers, one for the acoustic speech stream and one for the visual lip shape stream. It should be recalled that HMM can be defined as a type of NN system. The block diagram of the classifier used is illustrated in Figure 4.11. Design keys in this data fusion scheme are the selections of design methods for the audio HMM, the visual HMM, and the stream weight that is used for combining the outputs of the two different modal HMM sub-classifiers. A conventional selection of the design methods for the audio-visual HMM sub-classifiers is the MLE method, and an immediate reasonable selection for the stream weight is the GPD method using the classification error loss. Through experimental evaluations in a speaker-independent isolated word recognition task, the authors demonstrated that using GPD training for all three trainable modules, i.e., the audio HMM sub-classifier, the visual HMM sub-classifier, and the stream weight, achieved a significant error reduction, up to 80%, over the classifier

in which two sub-classifiers were trained with the MLE method.

4.5 Concluding Remarks

We have reviewed the recent attempts of speech pattern recognition using neural networks. The high discriminative power of NN-based recognizers mainly originates from the discriminative training that is a standard design approach to NN-based pattern recognition. To give a systematic view of the discriminative training methods, we used the GPD formalism as a discussion basis. The basic operations of this discriminative training are: 1) to introduce a discriminant function that measures the degree to which an input pattern belongs to some class, 2) to emulate the Bayes decision rule by the misclassification measure, 3) to introduce a loss function that enables one to evaluate the recognition result of a given training sample, and 4) to optimize the trainable system parameters of the recognizer at hand by using the preset loss function. In the training procedure, the selection of the loss is crucial. Various selections have been examined so far, including the classification error loss, the squared error loss, and the cross entropy loss. From the viewpoint of the GPD formalism, we showed the directness of the classification error loss to the optimal, minimum error-rate classification or, in other words, the minimum classification error probability condition.

In the chapter, we also summarized the framework of the fusion of multiple classification decisions. As an approach to increasing the design robustness to unknown samples, decision fusion, especially the averaging scheme over sub-classifier decisions, has attracted the recent researchers' interests. There are several possibilities for the embodiments of the framework. We particularly focused on two cases: 1) designing multiple sub-classifiers over a single training set with a single method of feature representation, and 2) designing multiple sub-classifiers over a single training set with multiple methods of feature representation. The utility of the decision fusion approach has been clearly demonstrated in many experimental studies, but its theoretical analysis is still insufficient due to the lack of a mathematical framework for analyzing the statistical characteristics of class boundary estimates.

Following the conventional taxonomy, we have to distinguish between HMM and NN. However, we basically think that there are no significant differences in system structure between these two types of systems. Actually, both are embodiments of a wider concept of a graphical model. In this light, we also think that in classifier design, one should not adhere to empirical selections of system structure, such as a hybrid system that simply combines existing systems. This is why we have only briefly discussed the issue of NN's architecture selection. Needless to say, a proper design of NN structure is important. It should probably reflect the nature of patterns that need to be classified. In addition, training methods, such as discriminative training, should be further evolved so as to cover the determination of classifier structure

as well as the adjustment of preset system parameters.

In Section 4.2.5.7, we introduced the global scope design method that uses the chain rule of differential calculus. One may notice that the mechanism of the global training is equivalent in operation to the error back-propagation algorithm developed for MLP networks, though the mechanism shown is more general. By using the global design strategy, one can in principle design feature extractors that are better suited to classification than those determined in empirical ways. A future research effort will be to discover a feature extractor that models the salient nature of speech signals for classification.

As summarized above, the selection of the loss function is important, and also the classification error count loss possesses the special feature of directness to the minimum error-rate classification. Further analyses on this loss selection are clearly desired for advancing pattern recognition methodology as well as NN-based speech recognition technology.

References

[1] S. Amari; "A theory of adaptive pattern classifiers," IEEE Trans. EC, vol. EC-16, pp. 299-307 (1967 3).

[2] P. Clarkson and P. Moreno; "On the use of support vector machines for phonetic classification," Proc. ICASSP99, paper no.-2104 (1999 5).

[3] A. Duchon and S. Katagiri; "Increasing the Robustness of GPD-Based Algorithms," Proc. ASJ Spring Conference, pp. 205-206 (1992 3).

[4] R. Duda and P. Hart; "Pattern Classification and Scene Analysis," New York: Wiley (1973).

[5] K. R. Farrell; "Model Combination and Weight Selection Criteria for Speaker Verification," Neural Networks for Signal Processing IX, IEEE, pp. 439-448 (1999).

[6] K. Farrell; "Networks for Speaker Recognition," in "Handbook of Neural Networks for Speech Processing (ed. S. Katagiri)", Artech House (2000).

[7] J. Fritsch, H. Hild, U. Meier, and A. Waibel; "Time-Delay Neural Networks and NN/HMM Hybrids: A Family of Connectionist Continuous-Speech Recognition Systems," in "Handbook of Neural Networks for Speech Processing (Ed. S. Katagiri)," pp. 245-280 Boston: Artech House (2000).

[8] P. Haffner, M. Franzini, and A. Waibel; "Integrating time alignment and neural networks for high performance continuous speech recognition," Proc. ICASSP91, pp. 105-108 (1991).

[9] H. Iwamida, S. Katagiri, E. McDermott, and Y. Tohkura; "A Hybrid Speech Recognition System Using HMMs with an LVQ-Trained Codebook," J. Acoust. Soc. Jpn. (E), Vol. 11, No. 5, pp. 277-286 (1990 9).

[10] B.-H. Juang and S. Katagiri; "Discriminative learning for minimum error classification," IEEE, Trans. SP., vol. 40, no. 12, pp. 3043-3054 (1992 12).

[11] B.-H. Juang, W. Chou, and C.-H. Lee; "Minimum classification error rate methods for speech recognition," IEEE Trans. SAP, vol. 5, pp. 257-265 (1997 3).

[12] S. Katagiri, C.-H. Lee, and B.-H. Juang; "New discriminative training algorithms based on the generalized probabilistic descent method," IEEE, Neural Networks for Signal Processing, pp. 299-308 (1991 9).

[13] S. Katagiri and C.-H. Lee; "A new hybrid algorithm for speech recognition based on HMM segmentation and learning vector quantization," IEEE Trans. SAP, vol. 1, pp. 421-430 (1993 4).

[14] S. Katagiri, B.-H. Juang, and A. Biem; "Discriminative feature extraction," in Artificial Neural Networks for Speech and Vision (R. Mammon Ed.), London, U.K.: Chapman and Hall, pp. 278-293 (1994).

[15] S. Katagiri, B.-H. Juang, and C.-H. Lee; "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," Proc. IEEE, vol. 86, no. 11, pp. 2345-2373 (1998 11).

[16] S. Katagiri (Ed.);"Handbook of Neural Networks for Speech Processing," Boston: Artech House (2000).

[17] S. Katagiri; "Minimum Classification Error Networks," in "Handbook of Neural Networks for Speech Processing (Ed. S. Katagiri)," pp. 307-354 Boston: Artech House (2000).

[18] T. Kohonen, G. Barna, and R. Chrisley; "Statistical pattern recognition with neural networks: benchmarking studies," Proc. of ICNN, vol. 1, pp. I-61- I-68 (1988 7).

[19] T. Kohonen; "Self-Organizing Feature Maps," New York: Springer-Verlag (1995).

[20] E. McDermott and S. Katagiri; "LVQ-based shift-tolerant phoneme recognition," IEEE Trans. SP, vol. 39, pp. 1398-1411 (1991 6).

[21] C. Miyajima, K. Tokuda, and T. Kitamura; "Audio-visual Speech Recognition Using Minimum Classification Error Training," in "Neural Networks for Signal Processing X," IEEE (2000).

[22] L. Rabiner and B.-H. Juang; "Fundamentals of Speech Recognition," Englewood Cliffs: Prentice-Hall (1993).

[23] A. J. Robinson; "An application of recurrent nets to phone probability estimation," IEEE Trans. NN, vol. 5, no. 2, pp. 298-305 (1994 3).

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams; "Learning Internal Representations by Error Propagation," in "Parallel Distributed Processing: Explorations in the Microstructure of Cognition (D. E. Rumelhart et al., Eds.)," MIT Press (1986).

[25] M. Schuster and K. K. Paliwal; "Bidirectional recurrent neural networks," IEEE Trans. SP, vol. 45, no. 11, pp. 2673-2681 (1997 11).

[26] V. Tresp; "Committee Machines," in "Handbook of Neural Network Signal Processing (eds. Y.-H. Hu and J.-N. Hwang)," Boca Raton: CRC Press (2002).

[27] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang; "Phoneme recognition using time-delay neural networks," IEEE Trans. ASSP, vol. 37, pp. 328-339 (1989 3).

4.6 Appendix: Maximizing Mutual Information

There is another possible definition of the loss that is related to the cross entropy loss (4.22). This has been used as a loss for the discriminative training of HMM speech classifiers, but to the best of the authors' knowledge, it has not been used for the design of NN-based speech classifiers. For reference purposes, we introduce this alternative choice.

The loss is referred to as mutual information loss, and it is defined as

$$I_k(X;\Lambda) = \ln \frac{p_\Lambda(X|C_k)}{\sum_j^M p_\Lambda(X|C_j)P(C_j)}, \tag{4.25}$$

where it is assumed that a training sample X belongs to C_k. Training using this measure aims at correct classification by maximizing the mutual information over the possible classes, or in other words, increasing the separability of the classes. Clearly, this training follows the discriminative training concept.

For discussion purposes, we consider the negative mutual information and reach the following inequality through simple rewriting operations:

$$\begin{aligned} -I_k(X;\Lambda) &= \ln \left\{ P(C_k) + \frac{\sum_{j,j\neq k}^M p_\Lambda(X|C_j)P(C_j)}{p_\Lambda(X|C_k)} \right\} \\ &\geq -\ln p_\Lambda(X|C_k) + \ln \left\{ \sum_{j,j\neq k}^M P(C_j) e^{\ln p_\Lambda(X|C_j)} \right\}. \end{aligned} \tag{4.26}$$

Here, assuming the logarithmic likelihood, $\ln p_\Lambda(X|C_k)$, to be the discriminant function, one can treat the bottom line expression of (4.26) as a kind of misclassification measure:

$$d_k(X;\Lambda) = -g_k(X;\Lambda) + \ln\left\{\sum_{j,j\neq k}^{M} P(C_j)e^{g_j(X;\Lambda)}\right\}. \tag{4.27}$$

Then, the inequality,

$$-I_k(X;\Lambda) \geq d_k(X;\Lambda) \tag{4.28}$$

holds true. Clearly, maximizing the mutual information leads at least to minimizing the misclassification measure (4.28). Consequently, training based on the maximization of the mutual information is considered discriminative training that uses the linear loss and the misclassification measure (4.27).

Like the case of the squared error loss, this training is certainly a type of discriminative training. However, due to the discrepancy between the smoothed error count loss and the linear loss used here, this training cannot guarantee that the minimum classification error condition will be achieved.

5

Large Vocabulary Speech Recognition Based on Statistical Methods

Jean-Luc Gauvain and Lori Lamel
LIMSI, France

CONTENTS

5.1 Introduction

Speech recognition is concerned with converting the speech waveform, an acoustic signal, into a sequence of words. Today's most practical approaches are based on a statistical modelization of the speech signal. This chapter provides an overview of the main topics addressed in large vocabulary speech recognition, that is: language modeling, lexical representation, acoustic-phonetic modeling and decoding. For over a decade large vocabulary, continuous speech recognition has been one of the focal areas of research in speech recognition, serving as a test bed to evaluate models and algorithms. This chapter focuses on the statistical methods used in state-of-the-art speaker-independent, large vocabulary continuous speech recognition (LVCSR). The reader will notice that although this chapter is dedicated to data driven statistical modeling of speech, prior knowledge about speech, and language is also taken into account, such as for example the assumption that words can be coded by a phonemic representation. Some of the primary application areas for LVCSR technology are dictation, spoken language dialog, and transcription systems for information retrieval from spoken documents.

5.2 Overview

From a statistical point of view, speech is assumed to be generated by a language model which provides estimates of $\Pr(W)$ for all possible word strings $W = (w_1, w_2, ...)$, and an acoustic model represented by a probability density function $f(X|W)$ encoding the message W in the signal X. The goal of speech recognition is generally defined as finding the most likely word sequence given the observed acoustic signal, i.e., of maximizing the probability of W given the speech signal X, or equivalently, maximizing the product $\Pr(W)f(X|W)$.

LVCSR systems use acoustic units corresponding to phones or phones-in-context, * where each word is described by one or more phone transcriptions. Assuming that the speech signal X depends only on the underlying phone sequence $H = (h_1, h_2, ...)$, then $f(X|W)$ can be rewritten as $\sum_H \Pr(H|W)f(X|H)$ where the summation is taken over the set pronunciations corresponding to the word sequence W. In practice this set is reasonably small as the average number of pronunciation variants per word is less than two. The underlying speech generation model is illustrated in Figure 5.1. The word sequence produced by the language model is successively transformed by two transducers, the pronunciation model and the acoustic model, to yield the speech signal.

This formulation of the LVCSR problem leads to the following four main considerations:

- The language modeling problem, i.e., computing the a priori probability $\Pr(W)$. It is usually estimated from relative *n*-gram frequencies in transcriptions of speech data as well as related text corpora.

- The pronunciation modeling problem, i.e., the computation of $\Pr(H|W)$. This relies on a pronunciation dictionary which may include estimates of the word pronunciation probabilities.

- The acoustic modeling problem, i.e., determining the structure of the probability density function $f(X|H)$ and estimating its statistical parameters from speech samples. The most predominant approach uses continuous density hidden Markov models (HMM) to represent context-dependent phones.

- The search problem, i.e., determining the best word hypothesis for the speech data given the models. This is a big challenge for LVCSR due to the large vocabulary and language model size.

*In this chapter the term phone is used to refer to acoustic units without attempting to label them as phonemic (referring to the elementary and distinctive sounds in the language) or phonetic (the observed realization of the elementary sounds). Contextual phone units (phones-in-context) implicitly model what can be considered allophones, i.e., contextual phonetic variants of the underlying phoneme.

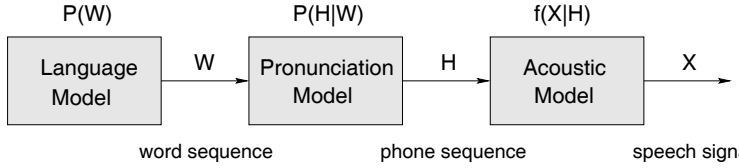

FIGURE 5.1
LVCSR speech generation model: The word sequence W produced by the language model is successively transformed by the pronunciation model ($P(H|W)$) and the acoustic model ($f(X|H,W)$), resulting in the speech signal X.

The principles on which most state-of-the-art LVCSR systems are based have been known for many years now, and include the application of the communication theory to speech recognition [7, 56, 57], the use of a spectral representation of the speech signal [26, 27], the use of dynamic programming for decoding [130, 131], and the use of context-dependent acoustic models [19, 76, 119]. Despite the fact that some of these techniques were proposed well over 15 years ago, considerable progress has been made in recent years in part due to the availability of large speech and text corpora, and improved processing power which have allowed more complex models and algorithms to be implemented.

The main components of a generic speech recognition system are shown in Figure 5.2 along with the requisite knowledge sources (speech and textual training materials and the pronunciation lexicon) and the main training and decoding processes. The acoustic and language models resulting from the training procedure are used as knowledge sources during decoding, after feature analysis has been carried out by the acoustic front-end. The remainder of this chapter is devoted to discussing these main constituents and knowledge sources. Some indicative performance levels are provided for three representative LVCSR tasks, and issues concerning language portability are discussed.

5.3 Language Modeling

Language models (LMs) capture regularities in spoken language and are used in speech recognition to estimate the probability of word sequences. While grammatical constraints described by hand-crafted context-free grammars have been used for small to medium size vocabulary tasks, LVCSR is essentially always based on data driven approaches. The most popular statistical method is the so called *n*-gram model, which attempts to capture the syntactic and semantic constraints of the language by estimating the frequencies of sequences of n words. The assumption is made that the probability of a given word string $W = (w_1, w_2, ..., w_k)$ can be ap-

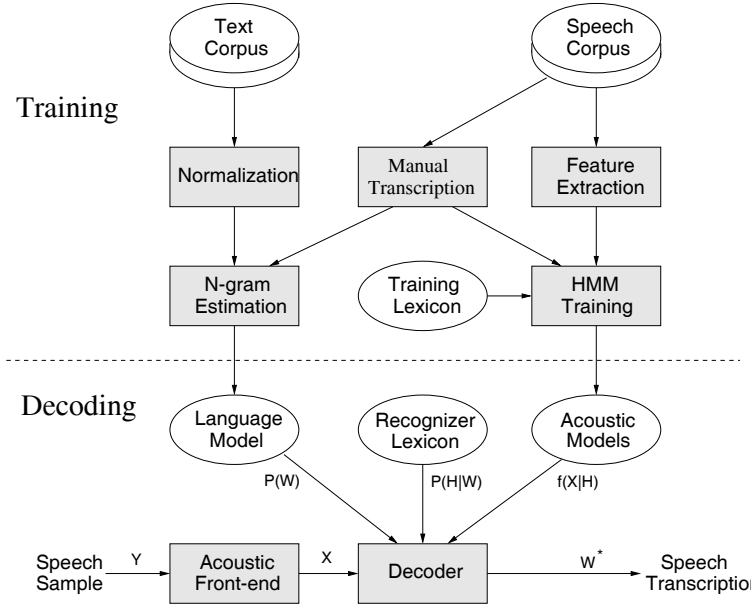

FIGURE 5.2
System diagram of a generic speech recognizer based on statistical models, including training and decoding processes and the main knowledge sources.

proximated by the following forward sequential decomposition

$$P(W) = \prod_{i=1}^{k} \Pr(w_i | w_{i-n+1}, ..., w_{i-2}, w_{i-1})$$

thereby reducing the word history to the preceding $n-1$ words. It should be noted that other decompositions of $P(W)$ can also be appropriate, for example, a backward decomposition will lead to a backward n-gram model.

A prerequisite for estimating n-gram language models is the availability of appropriately processed text corpora. As can be seen in Figure 5.2, language models are usually estimated from manual transcriptions of speech corpora and from normalized text corpora. To ensure accurate models, the texts need to be as representative as possible of the expected audio input to be transcribed. Text preparation entails locating appropriate sources of text data and audio transcriptions, and processing them in a homogeneous manner. Language models are generally optimized and compared by measuring the perplexity of a set of left out data, referred to as LM development data. This so-called test set perplexity of the language model M is defined as:

$$\text{Px}(T|M) = P(T|M)^{-\frac{1}{L}} \simeq (\prod_{i=1}^{L} P(w_i | w_{i-2}, w_{i-1}))^{-\frac{1}{L}}$$

for a given text $T = (w_1, ..., w_L)$ and a trigram LM (i.e., $n = 3$). $P(T|M)$ denotes the language model estimate of the text probability. The perplexity depends on both the language being modeled and the model, i.e., it gives a combined estimate of how good the model is and how complex the language is [57]. If the left out data set is representative of the model, the perplexity can be seen as a measure of the average branching factor, i.e., the vocabulary size of a memoryless uniform language model with same entropy as the language model under consideration.

5.3.1 Text Preparation

Although ideal language model training data would consist of large corpora of transcribed audio data representative of the targeted task, in practice such data are difficult to obtain. Therefore a variety of other more or less closely related text materials are usually used for language model training.

Given a large text corpus it may seem relatively straightforward to construct *n*-gram language models. Most of the steps are pretty standard and make use of tools that count word sequence occurrences [20]. The main considerations are the choice of the vocabulary, the definition of words (treatment of compound words and acronyms), and the choice of the LM back-off strategy (cf. Section 5.3.3). There is, however, a significant amount of effort needed to process (or normalize) the texts before they can be used. One motivation for the normalization is to reduce lexical variability so as to increase the coverage for a fixed size task vocabulary. The processing decisions are generally language-specific.

Numerical expressions and dates are typically expanded to approximate the spoken form and to reduce the lexical variety ($150 → *one hundred fifty dollars*, 1991 → *nineteen ninety one* or *one thousand nine hundred and ninety one*). Some example transformations are shown in Figure 5.3 along with the rule probabilities. For example, the word *hundred* followed by a number $< nb >$ can be replaced by *hundred and* 50% of the time; and 15% of the sequence *million dollars* are replaced with just the word *million* [38].

HUNDRED <nb>	$\Longrightarrow$	HUNDRED AND <nb> (0.50)
ONE EIGHTH	$\Longrightarrow$	AN EIGHTH (0.50)
CORPORATION	$\Longrightarrow$	CORP. (0.29)
INCORPORATED	$\Longrightarrow$	INC. (0.22)
ONE HUNDRED	$\Longrightarrow$	A HUNDRED (0.19)
MILLION DOLLARS	$\Longrightarrow$	MILLION (0.15)
BILLION DOLLARS	$\Longrightarrow$	BILLION (0.15)

FIGURE 5.3
Some example transformation rules applied during text normalization with associated probabilities.

Further semi-automatic processing is necessary to correct frequent errors inherent in the texts (such as obvious misspellings *million*, *officials*) or arising from processing with the distributed text processing tools. Some normalizations can be considered as "decompounding" rules in that they modify the word boundaries and the total number of words. These concern the processing of ambiguous punctuation markers (such as hyphen and apostrophe), the processing of digit strings, and treatment of abbreviations and acronyms (ABCD $\rightarrow$ A. B. C. D.). In agglutinative languages such as German, decompounding rules can be used to reduce the lexical variety. For example, the year 1991 which is written in standard German as *neunzehnhunderteinundneunzig*, can be transformed into the word sequence *neunzehn hundert ein und neunzig*. Depending upon the target application, the recognizer hypotheses may need to be mapped to a more appropriate written form. Other normalizations (such as sentence initial capitalization and case distinction) keep the total number of words unchanged, but reduce graphemic variability. In general a compromise is made between producing an output close to the standard written form of the language and the lexical coverage, with the final choice being largely application-driven.

5.3.2 Vocabulary Selection

Careful selection of the recognition vocabulary is important since on average, each out-of-vocabulary word causes more than one error (usually between 1.5 and 2 errors) [104]. The recognizer vocabulary is usually designed with the goal of maximizing lexical coverage for the expected input. A straightforward approach is to choose the N most frequent words in the training data which means that the usefulness of the vocabulary is highly dependent upon the representatively of the training data. To reduce this dependency it is common practice to select a word list suited to the expected test conditions by minimizing the system's out-of-vocabulary (OOV) rate on the LM development data. Therefore judicious selection of the development data is important. The best lexical coverage may be obtained by selecting the vocabulary using only a subset of the training data (such as the most recent data or data on a given topic) instead of using all the available data [16, 38]. An obvious way to reduce the error rate due to OOVs is to increase the size of the lexicon. Using a very large lexicon has been shown to improve performance, despite the potential of increased confusibility of the lexical entries [38].

5.3.3 N-gram Estimation

Using the maximum likelihood (ML) criterion, the n-gram probabilities are estimated from the frequencies of the word sequences of length n in the training corpus (texts or speech transcriptions). For example, the ML estimate of the trigram probability is given by:

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})}$$

where $C(\cdot)$ denotes the number of times the n-gram appears in the training data.

For large vocabulary sizes, many of the possible n-grams will not occur in even a very large training corpus. Due to the sparseness of the data, maximum likelihood estimates are clearly inadequate and need to be smoothed. Different approaches have been investigated to smooth the estimates of the probabilities of rare n-grams [17, 65]. The most common approach is to use a back-off mechanism [61] which relies on a lower order n-gram. If there is not enough data to obtain a robust estimate from the n-gram counts, a fraction of the probability mass is taken from the observed n-grams by discounting the ML estimates [45, 65, 140]. The probabilities of the rare n-grams are then estimated from the $(n-1)$-gram probabilities in a recursive manner as shown here for a trigram model:

$$\hat{P}(w_i|w_{i-2}, w_{i-1}) = \hat{P}(w_i|w_{i-1})\mathrm{B}(w_{i-2}, w_{i-1}),$$

where $\mathrm{B}(w_{i-2}, w_{i-1})$ is a back-off coefficient needed to ensure that the probability sum for a given context is equal to one. Computing the bigram estimate $\hat{P}(w_i|w_{i-1})$ follows the same principle. Backing-off offers an additional advantage in that the language model size can be arbitrarily reduced by increasing the cutoff frequencies below which the n-grams are not included in the model. This property can be used to reduce the amount of computational resources required during decoding. While 2-gram and 3-gram LMs are the most widely used, small improvements can be obtained with the use of longer span LMs such as 4-grams and 5-grams.

It is often the case that the LM training corpus is comprised of different sources of texts of different sizes and in different formats. Model interpolation is an easy way to combine training material from different sources. A language model is trained for each source and the resulting models are interpolated. The interpolation weights can be directly estimated on some development data with the EM algorithm. An alternative approach is to simply merge the n-gram counts and train a single language model on these counts. If some data sources are more representative than others for the task, the n-gram counts can be empirically weighted to minimize the perplexity on the development data set. While this can be effective, it has to be done by trial and error and cannot easily be optimized. In addition, weighting the n-gram counts can pose problems in properly estimating the back-off coefficients.

Word class or category-based language models can be used to reduce the dependency on the training data. Given some training data and a mapping which assigns each word to a unique category $\mathcal{C}(w)$, the training text can be tagged and the n-gram probabilities $\Pr(w_i|\mathcal{C}(w_{i-n+1}), ..., \mathcal{C}(w_{i-1}))$, which are often approximated by $\Pr(w_i|\mathcal{C}(w_i))\Pr(\mathcal{C}(w_i)|\mathcal{C}(w_{i-n+1}), ..., \mathcal{C}(w_{i-1}))$, can be estimated from the relative frequencies in the same manner as a regular word n-gram. The class assignment is often obtained by minimizing the perplexity of a bigram category model for a given number of word categories [64, 87]. It is also common practice to interpolate the category LM with the n-gram LM, in order to obtain a lower perplexity than that of the regular n-gram model. The resulting trigram probability estimates are:

$$P^*(w_i|w_{i-2}, w_{i-1}) = \alpha\hat{P}(w_i|w_{i-2}, w_{i-1}) + (1-\alpha)\hat{P}(w_i|\mathcal{C}(w_{i-2}), \mathcal{C}(w_{i-1})).$$

Other statistical language models have been investigated by mapping the word history $(w_1, ..., w_{i-1})$ onto equivalence classes other than the classical $(n-1)$-grams.

However, these modeling techniques such as decision tree models, maximum entropy models, or linguistically motivated models (probabilistic context-free and link grammars), have been used with moderate success leading to small gains over the much simpler n-gram model [115].

5.3.4 LM Adaptation

LVCSR systems use one or more language models, but these LMs are usually static, even though the choice of which model to use can be dynamic, dependent for example, on the dialog state. Language model adaptation is of interest for improving the model accuracy and for keeping the models up-to-date. Various approaches have been taken to adapt the language model based on the observed text so far, including the use of a *cache model* [58, 51], a *trigger model* [113], or *topic coherence modeling* [120]. The cache model is based on the idea that words appearing in a document will have an increased probability of appearing again in the same document. For short documents the number of words appearing is limited, and as a consequence the benefit is small. The trigger model attempts to overcome this increasing the probabilities of words that often co-occur with the trigger word when the trigger word is observed. In topic coherence modeling, selected keywords in the transcribed speech are used to retrieve articles on similar topics with which sublanguage models are constructed and used to rescore hypotheses. Despite the growing interest in adaptive language models, thus far only minimal improvements have been obtained compared to the use of very large, static n-gram models.

5.4 Pronunciation Modeling

The pronunciation dictionary is the link between the acoustic-level representation and the lexical items output by the speech recognizer. The accuracy of the acoustic models is partly dependent upon the consistency of the pronunciation dictionary.

Associated with each lexical entry are one or more pronunciations, described using the chosen elementary units (usually phonemes or phones). This set of units is evidently language dependent. For example, some commonly used phone set sizes are 45 for English, 50 for German and Italian, 35 for French and Mandarin (to which tones may be added), and 25 for Spanish. In generating pronunciation baseforms, most lexicons include standard full-form pronunciations and do not explicitly represent phonetic variants. This representation is chosen as most variants can be predicted by rules, and their use is optional. More importantly, there often is a continuum between different phonetic realizations of a given phoneme and the decision as to which occurred in any given utterance is subjective. By using a phone representation, no hard decision is imposed, and it is left to the acoustic models to represent the observed variants in the training data. While pronunciation lexicons are usually (at

Phone	*Example*	*Phone*	*Example*
Vowels		Fricatives	
i	b<u>ee</u>t	s	<u>s</u>ue
ɪ	b<u>i</u>t	z	<u>z</u>oo
e	b<u>ai</u>t	ʃ	<u>sh</u>oe
ɛ	b<u>E</u>t	ʒ	mea<u>s</u>ure
æ	b<u>a</u>t	f	<u>f</u>an
ʌ	b<u>u</u>t	v	<u>v</u>an
ɑ	b<u>o</u>tt	θ	<u>th</u>in
o	b<u>oa</u>t	Plosives	
u	b<u>oo</u>t	b	<u>b</u>et
ʊ	b<u>oo</u>k	d	<u>d</u>ebt
ɝ	b<u>ir</u>d	g	<u>g</u>et
Diphthongs		p	<u>p</u>et
ɑʲ	b<u>i</u>te	t	<u>t</u>at
ɔʲ	b<u>oy</u>	k	<u>c</u>at
ɑʷ	b<u>ou</u>t	Affricates	
Reduced Vowels		ʧ	<u>ch</u>eap
ə	<u>a</u>bout	ʤ	<u>j</u>eep
ɨ	dat<u>e</u>d	Nasals	
ɚ	butt<u>er</u>	m	<u>m</u>et
Semivowels		n	<u>n</u>et
l	<u>l</u>ed	ŋ	thi<u>ng</u>
r	<u>r</u>ed	Syllabics	
w	<u>w</u>ed	m̩	bott <u>om</u>
y	<u>y</u>et	n̩	butt<u>on</u>
h	<u>h</u>at	l̩	bott<u>le</u>

FIGURE 5.4
Set of 45 phone symbols for English with illustrative words, with the portion corresponding to the phone sound underlined.

least partially) created manually, several approaches to automatically learn and generate word pronunciations have been investigated. Such approaches, while promising, have to date given only small performance improvements even when trained on manual transcriptions [112].

Pronunciation variants can be observed for a variety of words. Alternative pronunciations are obviously needed for homographs (words spelled the same, but pronounced differently) which reflect different parts of speech (verb or noun) such as *excuse, record, moderate*. Some frequent affixes such as *anti-, bi-, multi-, -ization* can be pronounced with a diphthong (/ɑʲ/) or a short vowel (/ɪ/ or /ə/). The upper part of Figure 5.5 gives some example words with multiple pronunciations, and their associated probabilities. Using a set of allophone models (c.f. Section 5.5.2), the pronunciation probabilities are estimated by first aligning the reference word transcription

COUPON	kupɑn (0.63) kyupɑn (0.37)
ORGANIZATION	ɔrgənIzeʃən (0.93) ɔrgənɑʲzeʃən (0.07)
HUNDRED	hʌndʒ˞d (0.44) hʌndrəd (0.34)
	hʌnʒ˞d (0.18) hʌnrəd (0.04)
MODERATE	mɑdə˞It (0.82) mɑdə˞et (0.18)
TO	tə (0.66) tu (0.34)
I_DON'T_KNOW	ɑʲdonno (0.57) ɑʲdontno (0.05)
	ɑʲdʌno (0.28) ɑʲdno (0.10)
DON'T_KNOW	donno (0.73) dontno (0.18)
	dʌno (0.09)
DID_YOU	dIdyu (0.65)
	dIʤə (0.30) dIdyə (0.05)
GOING_TO	goɬŋtə (0.13) goɬŋtu (0.09)
	gʌnə (0.70) gcnə (0.08)

FIGURE 5.5

Some example lexical entries and their pronunciations along with estimate probabilities. For the compound words, the original concatenated pronunciation is given in the 1st line and the reduced forms are given in the 2nd line.

with the audio signal (using a lexicon containing equally likely alternative pronunciations), letting the Viterbi algorithm choose the best pronunciation for each word. The probabilities are then estimated from the relative frequencies of each variant.

Words of foreign origin, particularly proper names, may have different pronunciations depending upon the speaker's familiarity with the original language. It is also common for multisyllablic words to be pronounced with different numbers of syllables. For example, about 80% of the occurrences of *interest* and *conference*, and 20% of *company* are spoken with two syllables instead of three. If acoustic model training is carried out without allowing for appropriate pronunciation variants, there will necessarily be a misalignment of one or more phones, making the phone models less accurate. Experience has shown that careful lexical design improves speech recognition system performance [70].

In speech from fast speakers or speakers with relaxed speaking styles it is common to observe poorly articulated (or skipped) unstressed syllables, particularly in long words with sequences of unstressed syllables. Although such long words are typically well recognized, often a nearby function word is deleted. To reduce these kinds of errors, alternate pronunciations in the lexicon can allow schwa-deletion or syllabic consonants in unstressed syllables. Compound words have also been used as a way to represent reduced forms for common word sequences such as *don't know*, *did you*, and *going to*. Some of the reduced forms are so frequent that they have a commonly accepted written form (*gonna, dunno*). Some example compound words are shown in the lower part of Figure 5.5 along with estimates of the pronunciation probabilities for the different variants. These examples illustrate the interest in using compound words in recognition lexicons. Fluent speech effects can alternatively be

modeled using phonological rules [72, 101]. The principle behind the phonological rules is to modify the allowable phone sequences to take into account expected variations. These rules are optionally applied during training and recognition. Using phonological rules during training results in better acoustic models, as they are less "polluted" by wrong transcriptions. Their use during recognition reduces the number of mismatches. The same mechanism has been used to handle liaisons, mute-e, and final consonant cluster reduction for French.

As speech recognition research has moved from read speech to found audio data, the phone set has been expanded to include non-speech events. These can correspond to noises produced by the speaker (breath noise, coughing, sneezing, laughter, etc.) or can correspond to external sources (music, motor, tapping, etc).

5.5 Acoustic Modeling

One of the main challenges of acoustic modeling is to handle the variability present in the speech signal. Variability can arise from the linguistic context, or can be associated with the non-linguistic context such as the speaker (e.g., physical characteristics, speaking style, mood, etc.) and the acoustic environment (e.g., background noise, music) and recording channel (e.g., direct microphone, telephone). Most state-of-the-art LVCSR systems make use of hidden Markov models (HMMs) for acoustic modeling [110, 145], which consists of modeling the probability density function of a sequence of acoustic feature vectors. Other approaches include segment based models [43, 102, 152] and neural networks [2, 52] to estimate the acoustic observation likelihoods. With exception of the acoustic likelihood computation, all systems make use of the HMM framework to combine linguistic and acoustic information in a single network representing all possible sentences.

5.5.1 Acoustic Front-end

The first step of the acoustic feature analysis is digitization, or conversion of the continuous speech signal into discrete samples. The most commonly used sampling rates are 16kHz and 10kHz for direct microphone input, and 8kHz for telephone signals. The next step is feature extraction (also called front-end analysis), which has the goal of representing the audio signal in a more compact manner by trying to remove redundancy and reduce variability, while keeping the important linguistic information [53]. An inherent assumption is that although the speech signal is continually changing, due to physical constraints on the rate at which the articulators can move, the signal can be considered quasi-stationary for short periods (on the order of 10 to 20ms).

The most popular set of features are cepstrum coefficients obtained with a Mel Frequency Cepstral (MFC) analysis [21] or with a Perceptual Linear Prediction (PLP)

analysis [51]. Cepstral parameters are less correlated than direct spectral components, which simplifies estimation of the acoustic model parameters by reducing the need for modeling the dependency between features. In both cases a Mel scale short term power spectrum is estimated on a fixed window (usually in the range of 20 to 30ms). In order to avoid spurious high frequency components in the spectrum due to discontinuities caused by windowing the signal, it is common to use a tapered window such as a Hamming window. The window is then shifted, and the next feature vector computed. The most commonly used offset is 10ms. This acoustic parameterization converts the speech signal into a sequence of feature vectors X, each vector representing a 10ms interval referred to as a frame or a feature vector:

$$X = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T).$$

The Mel scale approximates the frequency resolution of the human auditory system, being linear in the low frequency range (below 1000 Hz) and logarithmic above 1000 Hz. The cepstral parameters are obtained by taking an inverse transform of the log of the filterbank parameters. In the case of the MFC coefficients, a cosine transform is applied to the log power spectrum, whereas a root-Linear Predictive Coding (LPC) analysis is used to obtain the PLP cepstrum coefficients. Both set of features have been used with success for LVCSR, but PLP analysis has been found to be slightly more robust in presence of background noise [63, 143].

Cepstral mean removal (subtraction of the mean from all input frames, generally sentence based) [29] is often used to reduce the dependency on the acoustic recording conditions. Computing the cepstral mean requires that all of the signal is available prior to processing, which is not the case for certain applications where processing needs to be synchronous with recording. In this case, a modified form of cepstral subtraction can be carried out where a running mean is computed from the N last frames (N is often on the order of 100, corresponding to 1s of speech). It is also common to normalize the feature variance, so that each resulting cepstral coefficient has a unity variance.

In order to capture the dynamic nature of the speech signal, the feature vector is usually augmented with "delta" parameters. The delta parameters are computed by taking the first and second differences of the features in successive frames. As a result a typical feature vector $\mathbf{x}_t$ will include 12 cepstrum coefficients plus the normalized log-energy, along with the first and second order derivatives, i.e., a total of 39 components. Instead of using these fixed delta features, linear discriminant transforms are sometimes used to better optimize the feature vector for the acoustic models [47, 68].

Vocal tract length normalization (VTLN), a technique which performs a simple speaker normalization at the front-end level [3], is also often used in LVCSR. The normalization consists of performing a frequency warping to account for differences in vocal track length, where the appropriate warping factor is chosen from a set of candidate values by maximizing the test data likelihood based on a first decoding pass transcription and some acoustic models [77]. VTLN must also be applied during the training process to obtain models suited to decode the normalized test data. This

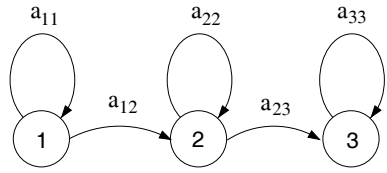

FIGURE 5.6
A simple 3-state left-to-right HMM topology commonly used for allophone modeling in LVCSR. The model generates at least 3 speech frames per allophone, resulting in a minimal phone segment duration of 30ms for frame rate of 100Hz.

normalization has been shown to give significant error rate reduction in particular on telephone conversational speech [128].

5.5.2 Modeling Allophones

Modeling allophones with Hidden Markov models is popular because these models work reasonably well, and their parameters can be efficiently estimated using well established techniques [110]. Allophone models offer a wide spectrum of contextual dependencies and back-off mechanisms to model rare contexts. The production of speech feature vectors is modeled in two steps. First, a small Markov chain is used to generate a sequence of states, and second, speech vectors are drawn using a probability density function (PDF) associated to each state. The Markov chain is described by the number of states and the transitions probabilities between states. While different model topologies have been proposed, most make use of left-to-right state sequences. The most commonly used configurations have 3 to 5 emitting states per allophone model, where the number of states imposes a minimal duration for the phone. Some configurations allow certain states to be skipped, thereby reducing the required minimal duration. The probability of an observation (i.e., a speech vector) is assumed to be dependent only on the current state.

Given an N-state HMM with parameter vector λ, the HMM stochastic process is described by the following joint probability density function of the observed signal $X = (\mathbf{x}_1, ..., \mathbf{x}_T)$ and the unobserved state sequence $S = (s_0, ..., s_T)$,

$$f(X, S|\lambda) = \pi_{s_0} \prod_{t=1}^{T} a_{s_{t-1}s_t} f(\mathbf{x}_t|s_t)$$

where π_i is the initial probability of state i, a_{ij} is the transition probability from state i to state j, and $f(\cdot|s)$ is the emitting PDF associated with each state s. Figure 5.6 shows the transition structure of a 3-state left-to-right HMM topology commonly used for allophone modeling in LVCSR.

The most frequently used state output PDF for speaker-independent systems is a

SISTER	/sIst3ʴ/
triphones:	s(*,I) I(s,s) s(I,t) t(s,3ʴ) 3ʴ(t,*)
quinphones:	s(*,Is) I(s,st) s(sI,t3ʴ) t(Is,3ʴ) 3ʴ(st,*)

FIGURE 5.7
Examples of allophonic transcriptions in terms of intra-word triphones and quinphones. Each contextual unit is defined by the central phone followed by its phone context shown in parentheses (left-context, right-context). * is a wildcard signifying any context.

mixture of Gaussians with 16 to 32 components,

$$f(\mathbf{x}_t|s) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mathbf{x}_t|\mathbf{m}_{sk}, \Sigma_{sk})$$

where $\mathbf{m}_{sk}$, Σ_{sk} and ω_k denote respectively the mean vector, the covariance matrix and the mixture weight of the k-th Gaussian component of state s. To reduce the number of parameters and the inherent estimation problem linked to full covariance matrices, the covariance matrices are usually assumed to be diagonal. (Recently it has been demonstrated that non-diagonal covariance matrices can be used while keeping the estimation problem manageable [33, 98].)

Phone based models offer the advantage that recognition lexicons can be described using the elementary units of the given language (c.f. 5.4), and thus can benefit from many linguistic studies. It is of course possible to perform speech recognition without using a phonemic lexicon, either by use of word models (as was the more commonly used approach 15 years ago) or a different mapping such as fenones (which are small data-driven acoustic units) [8]. Compared to word models, subword units reduce the number of parameters, enable cross word modeling and facilitate porting to new vocabularies. Fenones offer the additional advantage of automatic training, but lack the ability to include *a priori* linguistic knowledge.

A given HMM can represent a phone without consideration of its neighbors (context-independent model) or a phone in a particular context (allophone model). Various types of contexts have been investigated from a single phone context (right- or left-context), left and right-context (triphone), position-dependent triphones (cross-word and within word triphones), function word triphones, and quinphones [142]. The context may or may not include the position of the phone within the word (word-position dependent), and word-internal and cross-word contexts may be merged or considered as separate models. Different approaches are used to select the contextual units based on frequency of occurrence and clustering techniques. The optimal set of modeled contexts is usually the result of a tradeoff between resolution and robustness, and is highly dependent on the available training data. This optimization is generally done by minimizing the recognizer error rate on some development data.

Using contextual phone models can be seen as replacing the phone transcription as specified in the pronunciation dictionary by a transcription in terms of allophones.

Position: state-position, word-begin, word-end, monophone
General classes: vowel, consonant, continuant, sonorant, voiced-consonant, voiceless, fricative, strident, stop, nasal, semivowel, aspirated, anterior, high, coronal, slack, rounded, tense, retroflex, syllabic, fillers
Vowel classes: high-vowel, low-vowel, rounded-vowel, tense-vowel, reduced, diphthong, front-vowel, back-vowel, long-vowel, short-vowel, retroflex-vowel, diphthong-F2up, diphthong-F2down
Consonant classes: labial, dental, alveolar, palatal, velar, affricate
Individual phones: (see Figure 5.4)

FIGURE 5.8
Example questions used for decision tree clustering.

Figure 5.7 gives the triphone and quinphone transcriptions for the word SISTER using only word internal units, i.e., the allophonic transcription is independent of the word context. When using cross-word triphones, the models used for the first and last phone of each word (or the first and last two phones in the case of quinphones) depend on the word context making the decoding problem significantly more complex.

A powerful technique to keep the models trainable without sacrificing model resolution is to take advantage of the state similarity among different models of a given phone by tying the HMM state distributions. This basic idea is used in most current systems although there are slight differences in the implementation and in the naming of the resulting clustered states (*senones* [54], *genones* [24], *PELs* [11], *tied-states* [149]). In practice both agglomerative clustering and divisive clustering have been found to yield model sets with comparable performance. Divisive decision tree clustering is particularly interesting when there are a very large number of states to cluster since it is at the same time both faster and more robust than a bottom-up greedy algorithm, and therefore much easier to tune. In addition, HMM state tying based on decision tree clustering has the advantage of providing a means to build models for unseen contexts, i.e., those contexts that do not occur in the training data [55, 148]. The set of questions typically concern the phone position, the distinctive features (and identities) of the phone and the neighboring phones [95], as shown in Figure 5.8. The most frequently used questions for a large American English model set are given in Figure 5.9.

5.5.3 HMM Parameter Estimation

Acoustic model training consists of estimating the parameters of each HMM from the available training data. For Gaussian mixture HMMs, this requires estimating the means and covariance matrices, the mixture weights and the transition probabilities. If Λ is the parameter vector of the HMMs to be trained on some data X, the maxi-

Question	*Log likelihood gain*	*Question*	*Log likelihood gain*
vowel[+1]	6.3%	phone-r[+1]	2.2%
sonorant[+1]	5.5%	phone-H[+1]	2.1%
sonorant[-1]	3.8%	strident[+1]	1.9%
front-vowel[+1]	3.6%	phone-l	1.8%
semivowel[+1]	3.6%	nasal[-1]	1.7%
voiced-consonant[+1]	3.1%	vowel[-1]	1.6%
wordbody-pos[0]	2.5%	high-vowel[+1]	1.5%
nasal[+1]	2.3%	voiceless[-1]	1.5%
voiceless[+1]	2.2%	phone-n[+1]	1.5%
wordbegin-pos[0]	2.2%	phone-s[+1]1]	1.4%

FIGURE 5.9

The most frequently used decision tree questions for an American English broadcast news transcription system [40]. The [+1] and [-1] indicate that the question has been applied to the right or left context respectively, and [0] to the phone itself.

mum likelihood (ML) estimate is

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}}\, f(X|\Lambda, W)$$

where W is the reference transcription of X. ML estimation of the model parameters is usually done with the Expectation-Maximization (EM) algorithm [22] which is an iterative procedure starting with an initial estimate of the model parameters. At each iteration the HMM states are aligned to the training data utterances and the parameters are reestimated based on this alignment using the Baum-Welch reestimation formulas [12, 60, 79]. This algorithm guarantees that the likelihood of the training data increases at each iteration. In the alignment step a given speech frame can be assigned to multiple states (with probabilities summing to one) using the forward-backward algorithm or to a single state (with probability one) using the Viterbi algorithm. This second approach yields a slightly lower likelihood but in practice there is very little difference in accuracy especially when large amounts of data are available. It is important to note that the EM algorithm does not guaranty finding the true ML parameter values, and even when the true ML estimates are obtained they may not be the best ones for speech recognition. Therefore, some implementation details such as a proper initialization procedure and the use of constraints on the parameter values are quite important.

Since the goal of training is to find the best model to account for the observed data, the performance of the recognizer is critically dependent upon the representatively of the training data. Some methods to reduce this dependency are discussed below in the subsection on HMM adaptation. Speaker-independence is obtained by estimating the parameters of the acoustic models on large speech corpora containing data from a large speaker population. Since there are substantial differences in speech

from male and female talkers, it is common practice to use separate models for male and female speech in order to improve recognition performance. These differences can be attributed to anatomical differences (on average females have a shorter vocal tract length resulting in higher formant frequencies, as well as a higher fundamental frequency) and social ones (female voice is often "breathier" caused by incomplete closure of the vocal folds). The gender-dependent models are often obtained from speaker-independent seed models using Maximum *A Posteriori* estimators [39] (cf. the next section on HMM adaptation). The gender-dependent models can be further adapted to each specific speaker. Gender-dependent modeling is just one example of the family of adaptive training schemes which are particularly well-suited to heterogeneous training data (such as broadcast news recordings which include a wide variety of acoustical conditions, speaker types and speaking styles). Adaptive training makes use of HMM adaptation techniques after partitioning the training data according to acoustic conditions and speaker clusters.

Since an HMM is far from being the correct model of the observed data, and there is only a limited amount of data available to estimate its parameters, it can be advantageous to replace ML training with an alternative discriminative training scheme. Techniques for large-scale discriminative training of the acoustic models using the Maximum Mutual Information Estimation (MMIE) criterion in place of conventional ML estimation have been studied. It has been demonstrated that MMIE-based systems can lead to sizable word error rate reductions on the transcription of conversational telephone speech [108]. For a given training sequence X with transcription W (a training corpus is composed of many of such training sequences), the MMIE criterion with fixed language model, consists of maximizing the posterior probability of the word sequence, i.e.,

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}} \frac{f(X|W)P(W)}{\sum_{W'} f(X|W')P(W')}$$

where the summation in the denominator is taken over all possible word sequences. For LVCSR the calculation of the denominator terms is computationally expensive, so it is usually approximated by considering only the most likely word hypotheses given in the form of a word lattice (cf. Section 5.6.2 and Figure 5.10). For more details about discriminative training the reader is referred to [141].

5.5.4 HMM Adaptation

The performances of speech recognizers drop substantially when there is a mismatch between training and testing conditions. Several techniques can be used to minimize the effects of such a mismatch, so as to achieve a recognition accuracy as close as possible to that obtainable under matched conditions. Acoustic model adaptation can be used to compensate mismatches between the training and testing conditions, such as those arising from differences in the acoustic environment, microphones, and transmission channels, or to improve model accuracy based on the observed test data for a particular speaker. When no prior knowledge of either the channel type,

the background noise characteristics or the speaker is available, adaptation has to be performed using only the test data in an unsupervised manner.

Four commonly used schemes to adapt the parameters of a speech HMM can be distinguished: Bayesian adaptation [39]; adaptation based on linear transformations [78]; data clustering based adaptation [32, 69]; and model composition techniques [31].

Bayesian estimation, also called MAP estimation, can be seen as a way to incorporate prior knowledge into the training procedure by adding probabilistic constraints on the model parameters. The difference between MAP training and standard ML training lies in the assumption of an appropriate prior distribution of the parameters to be estimated. If Λ is the parameter vector of the HMM to be trained on some data X with a transcription W, and if g is the prior PDF of Λ, then the MAP estimate $\breve{\Lambda}$ is defined as the mode of the posterior PDF of Λ, i.e.,

$$\breve{\Lambda} = \underset{\Lambda}{\operatorname{argmax}}\, f(X|\Lambda, W)g(\Lambda).$$

The HMM parameters are still estimated with the EM algorithm but using the MAP reestimation formulas [39]. This leads to the MAP adaptation technique where constraints on the HMM parameters are estimated based on the parameters of an existing model. Speaker-independent acoustic models can serve as seed models for gender or speaker adaptation using the gender/speaker specific data, i.e., $g(\Lambda)$ in the above equation is replaced by $g(\Lambda|\hat{\Lambda})$ where $\hat{\Lambda}$ is the parameter vector of the seed models. MAP adaptation can be used to adapt the models to any desired condition for which sufficient labeled training data are available. MAP estimation has the same asymptotic properties as ML estimation but when independent priors are used for different phone models the adaptation rate may be very slow, particularly for large models. It is therefore advantageous to represent correlations between model parameters in the form of joint prior distributions [121, 151].

Linear transforms are powerful tools for performing unsupervised speaker and environmental adaptation. The ML linear regression (MLLR) technique [25, 78] is particularly well-suited to unsupervised adaptation. Since the number of transformation parameters is small, it is possible to adapt large models with small amounts of data. It consists of finding the transformation $\breve{R}$ (usually an affine transformation of the HMM Gaussian means, $\breve{\mathbf{m}}_k = \mathbf{A}\mathbf{m}_k + \mathbf{b}$) which maximizes the likelihood of the adaptation data X for a given hypothesized transcription $\hat{W}$, i.e.,

$$\breve{R} = \underset{R}{\operatorname{argmax}}\, f(X|R, \hat{\Lambda}, \hat{W}).$$

The transform parameters $\mathbf{A}$ and $\mathbf{b}$ are shared by the different phone units and are therefore robust to recognition errors. To obtain the ML asymptotic properties it is necessary to use multiple linear transforms and to adjust the number of linear transformations to the amount of available adaptation data. This can be done efficiently by arranging the mixture components into a tree and dynamically defining the regression classes. In addition to the Gaussian means, MLLR adaptation is often applied to the variance parameters. This adaptation procedure can be applied to both the

test data and training data. A natural extension of this approach, speaker adaptive training (SAT) incorporates supervised MLLR in the training procedure and jointly estimates the training speaker MLLR transforms and the HMM parameters [4]. The resulting SAT models are better suited to MLLR speaker adaptation.

Given the small number of parameters for the MLLR transformation (on the order of 500 parameters for a single regression class with a block diagonal matrix), this adaptation technique is still suitable with as little as 20s of adaptation data (i.e., only about 2000 frames). If less data is available, other adaptation techniques using a smaller number of adaptation parameters are required. Data clustering based adaptation methods, such as the eigenvoices scheme [69] and the cluster adaptive training [32] are such techniques. They both use a weighted sum of canonical speaker cluster models to estimate the Gaussian mean vectors. These adaptation schemes can also be combined with standard MLLR and MAP adaptation.

Model composition is mostly used to compensate for additive noise by explicitly modeling the background noise (usually with a single Gaussian) and combining this model with the clean speech model [30]. For practical reasons, it is generally assumed that the noise density is Gaussian and that the noise corrupted speech model has the same structure and number of parameters as the clean speech model – typically a continuous density HMM with Gaussian mixture. Various techniques have been proposed to estimate the noisy speech models, including the log-normal approximation approach, a numerical integration approach, and a data driven approach [31]. Model composition has the advantage of directly modeling the noisy channel as opposed to applying blind adaptation techniques to the same problem.

5.6 Decoding

The LVCSR decoding problem is the design of an efficient search algorithm to deal with the huge search space obtained by combining the acoustic and language models. Strictly speaking, the aim of the decoder is to determine the most likely word sequence W^*, given the language model, the pronunciation dictionary and the acoustic models, i.e.,

$$W^* = \operatorname*{argmax}_W P(W|X) = \operatorname*{argmax}_W \sum_{H,S} P(W)P(H|W)f(X,S|W)$$

where the summation is taken over all possible pronunciations and all possible HMM state sequences corresponding to the word sequence W. In practice, however, it is common to search for the most likely HMM state sequence. This maximum approximation, also referred to as Viterbi search, leads to a simplified view of the decoding problem:

$$W^* \simeq \operatorname*{argmax}_W \max_{H,S} P(W)P(H|W)f(X,S|W).$$

This is an easier task, consisting of finding the best path through a trellis (the search space) where each node represents an HMM state at a given time. It has been shown that even though the Viterbi decoding gives only a crude approximation of the likelihood of the word sequence, the two word hypotheses are almost always very close. Some simple extensions of the Viterbi search are able to compensate for most of decoding approximations in particular to avoid penalizing words with many pronunciations.
In many speech recognition systems the first step of decoding is identifying the speech portions of the audio signal. This process is described in the next subsection, followed by more details on decoding strategies.

5.6.1 Speech/Non-speech Detection

Detecting portions of the audio signal containing speech is commonly referred to as speech detection or endpoint detection. A variety of approaches to endpoint detection have been proposed ranging from simple energy threshold based methods to methods requiring the extraction of more complex parameters such as pitch. A general view of the problem is one of data partitioning, which aims to divide a continuous audio stream into homogeneous acoustic segments. Partitioning consists of identifying speech and non-speech segments, and then clustering the speech segments, assigning metadata labels to each segment. The labels typically specify the signal bandwidth and gender, but can also specify the background characteristics and speaker identity. When transcribing inhomogeneous audio streams, partitioning the data prior to word recognition offers several advantages. First, in addition to the transcription of what was said, other interesting information can be extracted from the audio signal, such as the division into speaker turns and the speaker identities, and background acoustic conditions. Second, by clustering segments from the same speaker, acoustic model adaptation can be carried out on a per cluster basis, as opposed to on a single segment basis, thus providing more adaptation data. Third, prior segmentation can avoid problems caused by linguistic discontinuity at speaker changes. Fourth, by using acoustic models trained on particular acoustic conditions (such as wide-band or telephone band), overall performance can be significantly improved. Finally, eliminating non-speech segments and dividing the data into shorter segments (which can still be several minutes long), substantially reduces the computation time and simplifies decoding.
Various approaches have been proposed to partition a continuous stream of audio data. Most of these approaches rely on a two step procedure, where the audio stream is first segmented in order to locate acoustic changes which are assumed to be associated with changes in speaker, background or environmental condition, and channel condition. The segmentation procedures can be classified as being based on phone decoding [48, 81, 135], distance-based segmentations [67, 124], or on hypothesis testing [18, 136]. The resulting segments are then clustered (usually using Gaussian models), where each cluster is assumed to identify a speaker or more precisely, a speaker in a given acoustic condition. An alternative language-independent approach relies on an audio stream mixture model [37]. Each component audio source,

representing a speaker in a particular background and channel condition, is in turn modeled by a mixture of Gaussians. The segment boundaries and labels are jointly identified via an iterative maximum likelihood segmentation/clustering procedure using Gaussian mixture models and agglomerative clustering.

5.6.2 Decoding Strategies

Since it is often prohibitive to exhaustively search for the best path, techniques have been developed to reduce the computational load by limiting the search to a small part of the search space. Even for research purposes, where real-time recognition is not needed there is a limit on computing resources (memory and CPU time) above which the development process becomes too costly. The most commonly used approach for small and medium vocabulary sizes is the one-pass frame-synchronous Viterbi beam search [92] which relies on a dynamic programming algorithm. This basic strategy has been extended to deal with large vocabularies by adding features such as dynamic decoding [96], multi-pass search [91], and N-best rescoring [118].
Dynamic decoding can be combined with efficient pruning techniques in order to obtain a single pass decoder that can provide the answer using all the available information (i.e., that in the models) in a single forward decoding pass over of the speech signal. This kind of decoder such as the stack decoder [107] based on the A * algorithm or the one-pass frame synchronous dynamic network decoder [96], is very attractive for real-time applications.
Static decoders require much more memory than dynamic decoders when used with long span language models (3-gram or higher order), and as a consequence they are mostly used with smaller language models (usually 2-grams or constrained grammars). It has been recently shown that by proper optimization of a finite-state automaton[†] corresponding to a recognizer HMM network, substantial reduction of the overall network size can be obtained, enabling static decoding with long span LMs [90]. However, the size of the optimized network remains proportional to the LM size.
Multi-pass decoding can be used to progressively add knowledge sources in the decoding process, thus allowing the complexity of the individual decoding passes to be reduced and often resulting in a faster overall decoder [94]. For example, a first decoding pass can use a 2-gram language model and simple acoustic models, and later passes will make use of 3-gram and 4-gram language models with more complex acoustic models. This multiple pass paradigm requires a proper interface between passes in order to avoid losing information and engendering search errors. Information is usually transmitted via word lattices[‡] or word graphs (see Figure 5.10), although some systems use N-best hypotheses which are a list of the most likely

[†] An HMM-based speech recognizer can be seen as a transduction cascade which converts the observed feature vectors to a word string, where to some approximation, each transduction (phone model, word model or language model) can be represented as a finite-state automaton.

[‡] Lattices are graphs where nodes correspond to particular frames and where edges representing word hypothesis have associated acoustic and language model scores.

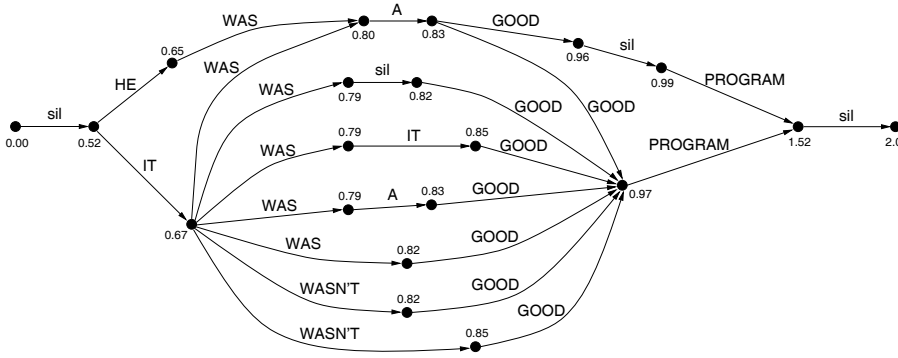

FIGURE 5.10

Example word lattice generated by a speech recognizer using a bigram language model for a 2.1s utterance. Each graph edge corresponds to a word hypothesis and a time interval (as specified by the time information on the nodes). In this example the word transcription with the highest likelihood is "sil IT WAS A GOOD PROGRAM sil" which happens to be what was said. (The acoustic and language model likelihoods are not given on the figure.)

word sequences with their respective scores. At the price of some acceptable approximations, word lattices (and N-best lists) can be generated with little overhead (about 10%) by modifying the bookkeeping of the partial hypotheses considered during regular decoding [100].

It can sometimes be difficult to add certain knowledge sources into the decoding process especially when they do not fit in the Markovian framework. This is the case when trying to use segmental information or to use grammatical information for long term agreement. Such information can be more easily integrated in a multi-pass system by rescoring the recognizer hypotheses after applying the additional knowledge sources. Evidently, the first pass used to generate the initial word lattice must be accurate enough to not introduce lattice errors which are unrecoverable with further processing.

In addition to multiple pass decoding, word lattices can be used to overcome the Viterbi approximation discussed above. As a matter of fact, true MAP decoding is a considerably easier task on a word lattice than on the original search space. Along the same lines, it has been proposed to use word lattices to perform a word based MAP decoding instead of word sequence MAP decoding, i.e., minimizing the word error instead of the word sequence (or sentence) error rate [83].

5.6.3 Efficiency

As discussed above, there are many efficient solutions to the search problem, however finding the optimal solution is always a trade-off between the model accuracy and efficient pruning. In general better models have more parameters, and therefore

require more computation. However since the models are more accurate, it is often possible to use a tighter pruning level (thus reducing the computational load) without any loss in accuracy.

Limitations on the available computational resources can significantly affect the design of the acoustic and language models, as for each operating point, the right balance between model complexity and pruning level must be found. Aggressive pruning is generally needed to achieve real-time operation for LVCSR tasks on currently available platforms. This inevitably is a source of search errors, and as such, many techniques have been proposed to reduce these search errors and to limit their effect on the recognizer accuracy. One of the most popular decoding strategies for real-time operation is the one-pass frame-synchronous dynamic network decoder which relies on a phonetic tree organization of the decoding network using LM state conditioned tree copies [5, 93, 96]. The success of such a single pass approach is highly dependent on the use of efficient pruning strategies associated with a language model lookahead [99, 117]. Multipass approaches can also be used successfully for close to real-time operation by chunking the data and running the different passes in parallel with a slight delay.

For speaker-independent LVCSR based on Gaussian mixture HMM, between 30 and 50% of the recognition time is spent in computing the HMM state likelihoods, with the remaining time corresponding to the search procedure itself. This is due to the large number of states needed to represent the context-dependent phone models, even when state tying is used. This computation can be reduced either by implementing a fast state likelihood computation which usually requires making some approximations, or by reducing the model size which has the additional advantage of reducing the memory requirements. A widely used technique for speeding up the state likelihood computation is vector quantization of the feature vector space in order to prepare a Gaussian short list for each HMM state and each region of the quantified feature space [13]. With this technique the number of Gaussian likelihoods to be computed during decoding for each input frame and each state can be reduced to a fraction of the number of Gaussians corresponding to the active states with only a small loss in accuracy.

Model and state tying are commonly used to improve the model accuracy but optimal tying (from the accuracy point of view) can still result in a very large model with 5 k to 30 k states when large amounts of training data are available. Parameter tying is also powerful technique to reduce the number of parameters, and can be applied to all the levels of the model structure (allophone model, state, and Gaussian) [127]. However, more flexibility is available for Gaussian PDF tying in that large model reductions can be obtained without sacrificing too much in terms of system accuracy. This is exemplified by the subspace distribution tying approach [84, 127], which in its most elementary implementation can be seen as a quantization of the model parameters.

The language model, usually a 3-gram or 4-gram back-off LM in state-of-the-art systems, can have a very large number of parameters (over 10 million), and therefore may require prohibitive amounts of memory. One of the attractive properties of n-gram models is the possibility of relying more on the back-off components by

increasing the cutoffs on the n-gram counts, thus reducing significantly the LM size (c.f. Section 5.3.3). More elaborate n-gram pruning techniques have also been proposed [123, 126] to substantially reduce the LM size with negligible loss in accuracy. An alternative approach to limit the memory requirements is to keep most of the LM parameters on the disk, since most n-grams are never used, combined with a cache of the scores for accessed LM states [111].

5.6.4 Confidence Measures

Confidence measures have been proposed as a way of detecting those hypothesized words that are likely to be erroneous by estimating word and sentence correctness [15, 42, 125, 138, 137]. At the sentence level the goal is to get an estimate of $\Pr(W|X)$ for the hypothesized word string W. One common approach consists of using the posterior $\Pr(W|X, \mathcal{M})$ as an estimate. This assumes that the recognizer models (acoustic model, language model, and lexicon designated by $\mathcal{M}$) are correct and that the decoder does not make any search errors. Further approximations may use simpler acoustic and language models to speed up the computation, for example, the word language model can be replaced by a phone language model [36]. For most LVCSR tasks, the concern is essentially for a word level confidence measure, i.e., the goal is to obtain an estimate of $\Pr(w_i|X)$ the posterior probability of the i-th word in the hypothesized word string, or alternatively $\Pr(w_i|X, \mathcal{M})$. An estimate of this latter probability can be efficiently computed by applying the Forward-Backward algorithm to a word graph generated by the speech recognizer [138]. However since this posterior probability relies on incorrect models, it is also common to use additional features such as word and phone durations, speaking rate, and signal-to-noise ratio to better approximate the word posterior probability $\Pr(w_i|X)$. All the predictors can be combined and mapped to the confidence score by using either a logistic regression [42], a generalized additive model [125], or a neural-network [137]. These models are trained on development data by maximizing a confidence score metric such the normalized cross entropy. The proper set of features depends on the particular application.

5.7 Indicative Performance Levels

This section provides some indicative measures of recognizer performance for a few LVCSR tasks, but makes no attempt to be exhaustive. Essentially all of todays state-of-the-art systems make use of the statistical modeling techniques presented in this chapter. Speech recognition technology has advanced greatly over the last decade. These advances can be clearly seen in the context of DARPA supported benchmark evaluations. This framework, known in the community as the DARPA evaluation paradigm, has provided the training materials (transcribed audio and textual cor-

pora for training acoustic and language models), test data and a common evaluation framework. In recent years the data have been provided by the Linguistics Data Consortium (LDC) and the evaluations organized by the National Institute of Standards and Technology (NIST) in collaboration with representatives from the participating sites and other government agencies. It is widely acknowledged that the performance of a speech recognizer is strongly dependent upon the task, which in turn is linked to the type of user, speaking style, environmental conditions, etc.

The commonly used metric for speech recognizer performance, the "word error" rate, is a measure of the average number of errors taking into account three error types with respect to a reference transcription: *substitutions* (the reference word is replaced by another word), *insertions* (a word is hypothesized that was not in the reference), and *deletions* (a word in the reference transcription is missed). The word error rate is defined as

$$\frac{\#\text{subs}+\#\text{ins}+\#\text{del}}{\#\text{ reference words}}$$

and is generally computed by aligning the reference and hypothesized transcriptions using a dynamic programming algorithm, where costs are associated with the different error types. Given this definition the word error can be more than 100%.

While this chapter addresses speech transcription (i.e., going from the audio signal to words), it should be kept in mind that additional information can be extracted from the audio signal. Extraction of some of this so-called "metadata", is discussed in Chapters 8 (Schwartz and Makhoul) and 12 (Allen). The metadata can be of an acoustic nature (speaker and gender information [73], audio type information [34, 122]) of a linguistic nature (case-sensitive texts, punctuation, named entities (names of persons, places, organizations), topics, or other semantic tags). The same HMM-based probabilistic framework has been used to assign tags [89, 134, 144]. Detailed semantic tagging is often required for dialog tasks where it is common to use task-dependent representations such as semantic frames, with predefined semantic slots and values.

5.7.1 Dictation

Dictation is the most obvious automatic speech recognition task, and has a long history of research and product development, resulting in low-cost, off-the-shelf systems for a variety of platforms and languages. While from the technological viewpoint, dictation is usually thought of as a "simple" transformation from speech to text, this view overlooks a variety of formatting and integration issues which are important for usability. Perhaps the most notable characteristic of the dictation task is that the speech data is produced with the explicit goal of being transcribed by a machine. The speech data in a dictation session comes from a single speaker and is recorded with a controlled signal acquisition setup. The linguistic content is usually somewhat limited and the word stream is quite close to the written form.

Although benchmarks of commercial dictation systems are not publicly available, dictation has served as a baseline performance measure in LVCSR, most notably in the benchmark tests sponsored by the U.S. DARPA programs and coordinated by

NIST. The close relationship between system development and evaluation (referred to as "assessment driven technology development") has led to large performance improvements in spite of increasing task difficulty. For read speech, the state-of-the-art in speaker-independent continuous speech recognition is exemplified by the last benchmark tests (1995/1996) on North American Business News task [104, 105]. The acoustic training data was comprised of about 160 h of read newspaper texts from several hundred speakers, and the language model training material was comprised of 400 M words of newspaper texts, from a variety of sources. On test data recorded with a close-talking microphone with an SNR of about 30 dB, word error rates around 7% were obtained using a 65 k word vocabulary. [§] The same read speech recorded with a table-top microphone in a computer room/office environment (noise level 55 dBA, SNR about 15 dB), resulted in a word error of about 14% with noise compensation. Without noise compensation the word error rates of systems trained on only clean speech data were over 50%. The word error for read newspaper texts recorded over long distance telephone lines was over 20%. Spontaneous dictation of business and financial news was addressed by asking subjects with experience in journalism to read about a subject and then dictate a text. The journalists were not allowed to read from a draft, but were allowed to reject ill-formed sentences [66]. The word error on this data was about 14%. Another task addressed speech recognition of non-native talkers. With a set of 40 adaptation sentences, speaker adaptation reduced the word error rate by a factor of two (from 21% to 11%). Although not an official benchmark result, comparable word error reductions have been obtained for native speakers on other tasks.

While the results given here are for American English, somewhat comparable results have been reported by various sites for other languages [71]. The LRE SQALE (Speech recognizer Quality Assessment for Linguistic Engineering) project [146], which aimed to assess language-dependent issues in multilingual recognizer evaluation, demonstrated that the same recognition technology and evaluation methodology used for American English could be successfully applied to a dictation task in British English, French, and German.

5.7.2 Speech Recognition for Dialog Systems

The speech recognizer is often considered a critical component of spoken dialog systems, which aim to enable vocal access to stored information. In order to provide user-friendly interaction with a machine, it is necessary to be able to recognize naturally spoken utterances from unknown speakers. In general each user interacts only briefly with the machine, so there is very little data available for model adaptation. Telephone services are a natural area for spoken dialog systems as the only means of interaction with the machine are via voice and have thus been the focus of many development efforts. Since all interaction with the caller is by speech, dialog design

[§] With the exception of the telephone recordings, the speakers were allowed to repeat their recording if unsatisfied with it. [66, 104]

and response generation are of particular importance in the context of natural, mixed-initiative dialog. Growing in popularity are information kiosks [35] and multimedia web interfaces, in which different modalities (tactile and audio) can be used for input and output. The speech recognizers of dialog systems are typically faced with more challenging acoustic conditions than for dictation tasks, being subject to channel distortions, varied handsets and noisy background conditions. The capability of the user to interrupt the machine is often considered crucial for usability.

In contrast to dictation applications where it is relatively straight-forward to obtain large written corpora for language modeling, for dialog systems it is usually necessary to collect application-specific data, which can represent a significant portion of the development effort [75]. Acquiring sufficient amounts of LM training data is more challenging than obtaining acoustic data. With 10 k queries relatively robust acoustic models can be trained, but this number of queries will typically contain fewer than 100 k words, which may be insufficient for word list development and for training *n*-gram language models. Also the queries are unlikely to yield a complete coverage of the task.

The most widely known efforts in evaluation of SLDSs are the DARPA ATIS task [50, 82, 109], the German national Verbmobil project [132], and the EC Language Engineering projects [85, 86]. A wide range of word error rates have been reported for the speech recognition components of spoken dialog systems, ranging from under 5% for simple travel information tasks using close-talking microphones to over 25% for telephone-based information retrieval systems. It is quite difficult to compare results across systems and tasks as different transcription conventions and text normalizations are often used. It should be noted that reporting word error rates can be somewhat misleading, since all differences between the *exact* orthographic form of the query and the recognizer output are counted as errors, and some of recognition errors (such as gender or plurals) are not important for understanding. A more appropriate measure could be the error rate on meaningful words or concepts used in later processing stages. In the DARPA ATIS benchmark tests [103, 104] the understanding error based on the spoken input was not much larger than the natural language understanding error obtained using manual orthographic transcriptions. In the case of multimodal systems, the effectiveness of speech must be assessed in coordination with the other modalities.

5.7.3 Transcription for Audio Indexation

A more recent application area is the transcription of general audio data, such as radio and television broadcasts,[¶] or meetings and teleconferences. Automatic speech recognition is a key technology for audio and video indexing and any kind of audio data mining. Several characteristics of this type of audio data can be noted. First, it can be considered "found" data in that it is produced for other reasons. To be able

[¶] The earliest work in this area that we are aware of is the NSF INFORMEDIA project [49] under the Digital Libraries News-on-Demand action line. A special section of the Communications of the ACM was recently devoted to this topic [88].

to automatically structure the data for other uses is only a secondary benefit. Using test data taken from a real task, as opposed to data recorded for evaluation purposes represents a major step for the community. Secondly, the data consists of a continuous audio stream, where there are multiple speaker turns (maybe overlapping), and there is no a priori segmentation into sentences. Thirdly, the signal capture and background environment can be only more or less controlled.

Two principle types of problems are encountered in automatically transcribing audio data streams: those relating to the varied acoustic properties of the signal, and those related to the linguistic properties of the speech. Noise robustness is also needed in order to achieve acceptable performance levels. In order to be robust with respect to the varied acoustic conditions, the acoustic models are typically trained on large corpora (several tens of hours to over a hundred hours) containing a variety of data types. The linguistic models are similarly trained on large text corpora from various sources with different linguistic properties, such as newspaper and newswire texts, Internet data, commercial transcriptions, and detailed transcriptions of acoustic data. Given the spontaneous nature of parts of the audio data, it is important to explicitly model extralinguistic phenomena such as filler words and breath noise.

State-of-the-art transcription systems (trained on 100h of acoustic data and over 200 M words of commercial transcripts) achieve word error rates of around 20% on unrestricted broadcast news data. Transcription performance varies quite a bit across the data types. The average word error rate reported on prepared, announcer speech is about 8% in the DARPA benchmark test data, but under 2% for some speakers. Performance decreases substantially for spontaneous portions (average word error 15%), degraded acoustic conditions (average word error 16%), or speech from non-native speakers (average word error over 25%). The transcription of broadcast data has also been a recent focus of research efforts in several other languages, including French, German, Italian, Japanese, Mandarin, and Spanish [14, 59, 62, 97, 106] using the same technology. The reported error for these languages are somewhat higher than for American English which can be at least partially attributed to the smaller amounts of training data available in these languages, and in particular to the difficulty of obtaining commercial transcripts for language model estimation.

Substantially higher word error rates, above 30-40% have been reported for the transcription of telephone conversational speech [147] using the Switchboard [44] and multilinugal Callhome (Spanish, Arabic, Mandarin, Japanese, German) corpora. The Callhome data is particularly challenging to transcribe as the conversations are between two people that know each other, and speak in a familiar manner about subjects of common interest.

5.8 Portability and Language Dependencies

Statistically-based speech recognition technology has been successfully employed for a variety of tasks and languages. The porting of a LVCSR system to a new task or another language requires the availability of sufficient amounts of transcribed training data and involves substantial effort to construct the acoustic and language models, and to develop the recognition lexicon. Often, however, the necessary resources are not available and generating them can be long and expensive.

Recent efforts have been directed at developing generic recognition models and the use of unannotated data for training purposes, in an aim to reduce the reliance on manually annotated training corpora and reducing development costs [1]. Methods to improve generality of the models are under investigation, but the problem is far from being solved.

Although English has been the predominant language for the computer world, there has been a large growth in the amount of information available in electronic form in many of the world's languages. Building a recognizer for another language is not so different than building a recognizer for a new task, particularly for close languages. Language-dependent system components (such as the phone set, the need for pronunciation alternatives, or phonological rules) evidently must be changed. Other language dependent factors are related to the definition and acoustic confusibility of the words in the language (such as homophone, monophone, and compound word rates) and the word coverage of a given size recognition vocabulary. Taking into account language specificities can evidently improve recognition performance. For example, tonal languages such as Chinese may benefit from explicit modeling of pitch, which in turn may require modifications to the feature analysis used.

There are two predominant approaches for bootstrapping acoustic models for another language. The first is to use acoustic models from an existing recognizer and a pronunciation dictionary to segment manually annotated training data for the target language. If recognizers for several languages are available, the seed models can be selected by taking the closest model in one of the available language-specific sets. An alternative approach is to use a set of global acoustic models, that cover a wide number of phonemes [116, 133]. This approach offers the advantage of being able to use multilingual acoustic models to provide additional training data, which is particularly interesting when only limited amounts of data (< 10 hours) for the target language are available.

Minimizing the required training data (or determining how to optimally acquire such data) remains an outstanding challenge. Standard HMM training requires an alignment between the audio signal and the phone models, which usually relies on an orthographic transcription of the speech data and a good phonemic lexicon. The orthographic transcription is usually considered as ground truth, that is the word sequence that should be hypothesized by the speech recognizer when confronted with the same speech segment. One can imagine training acoustic models in a less supervised manner, in which related linguistic information about the audio sample can be

used in place of the manual transcriptions required for alignment by incorporating this information in a language model. This language model can be used with acoustic models developed for another task to automatically transcribe the task-specific training data. Although in the beginning the error rate on new data is likely to be rather high, this speech data can be used to retrain the models of the recognition system. An iterative procedure can successively refine the models and the transcription [62, 74, 150, 139]. This approach is particularly promising for the transcription of readily available audio sources such as radio and television news broadcasts, that can provide an essentially unlimited supply of acoustic training data.

References

[1] http://coretex.itc.it

[2] D. Abberley, D. Kirby, S. Renals and T. Robinson, *The THISL Broadcast News Retrieval System,* Proc. ESCA ETRW on Accessing Information in Spoken Audio, 14-19, Cambridge, U.K., April 1999.

[3] A. Andreoum, T. Kamm and J. Cohen, *Experiments in Vocal Tract Normalization,* Proc. CAIP Workshop: Frontiers in Speech Recognition II, 1994.

[4] T. Anastasakos, J. McDonough, R. Schwartz and J. Makhoul, *A Compact Model for Speaker Adaptation Training,* Proc. ICSLP'96, 1137-1140, Philadelphia, PA, October 1996.

[5] X. Aubert, *One Pass Cross Word Decoding for Large Vocabularies Based on a Lexical Tree Search Organization,* Proc. ESCA Eurospeech'99, **4**:1559-1562, Budapest, Hungary, September 1999.

[6] S. Austin, R. Schwartz and P. Placeway, *The Forward-Backward Search Strategy for Real-Time Speech Recognition,* Proc. IEEE ICASSP-91, 697-700, Toronto, Canada, May 1991.

[7] L.R. Bahl, J.K. Baker, P.S. Cohen, N.R. Dixon, F. Jelinek, R.L. Mercer and H.F. Silverman, *Preliminary results on the performance of a system for the automatic recognition of continuous speech,* Proc. IEEE ICASSP-76, Philadelphia, PA, April 1976.

[8] L.R. Bahl, P. Brown, P. de Souza, R.L. Mercer and M. Picheny, *Acoustic Markov Models used in the Tangora Speech Recognition System,* Proc. IEEE ICASSP-88 **1**:497-500, New York, NY, April 1988.

[9] L.R. Bahl, F. Jelinek and R.L. Mercer, *A Maximum Likelihood Approach to Continuous Speech Recognition,* IEEE Trans. Pattern Analysis & Machine Intelligence, **PAMI-5**(2):179-190, March 1983.

[10] L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, D. Nahamoo and M. Picheny, *A Fast Match for Continuous Speech Recognition Using Allophonic Models,* Proc. IEEE ICASSP-92, CA, **1**:17-21, San Francisco, CA, March 1992.

[11] J. Baker, J. Baker, P. Bamberg, K. Bishop, L. Gillick, V. Helman, Z. Huang, Y. Ito, S. Lowe, B. Peskin, R. Roth and F. Scattone, *Large Vocabulary Recognition of Wall Street Journal Sentences at Dragon Systems,* Proc. DARPA Speech & Natural Language Workshop, 387-392, Harriman, NY, February 1992.

[12] Baum, L.E., T. Petrie, G. Soules, and N. Weiss, *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*, Ann. Math. Stat.. **41**:164-171, 1970.

[13] E. Bocchieri, *Vector quantization for efficient computation of continuous density likelihoods,* Proc. IEEE ICASSP-93, **2**:692-695, Minneapolis, MN, May 1993.

[14] F. Brugnara, M. Cettolo, M. Federico and D. Giuliani, *A Baseline for the Transcription of Italian Broadcast News,* Proc. IEEE ICASSP-00, Istanbul, Turkey, June 2000.

[15] L. Chase, *Word and acoustic confidence annotation for large vocabulary speech recognition*, Proc. ESCA Eurospeech'97, 815-818, Rhodes, Greece, September 1997.

[16] L. Chase, R. Rosenberg, A. Hauptmann, M. Ravishankar, E. Thayer, P. Placeway, R. Weide and C. Lu, *Improvements in Language, Lexical and Phonetic Modeling in Sphinx-II,* Proc. ARPA Spoken Language Systems Technology Workshop, 60-65, Austin, TX, January 1995.

[17] S.F. Chen and J. Goodman, *An empirical study of smoothing techniques for language modeling,* Computer, Speech & Language, **13**(4):359-394, October 1999.

[18] S.S. Chen and P.S. Gopalakrishnan, *Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion*, Proc. DARPA Broadcast News Transcription & Understanding Workshop, 127-132, Landsdowne, VA, February 1998.

[19] Y.L. Chow, R. Schwartz, S. Roukos, O. Kimball, P. Price, F. Kubala, M.O. Dunham, M. Krasner and J. Makhoul, *The Role of Word-Dependent Coarticulatory Effects in a Phoneme-Based Speech Recognition System*, Proc. IEEE ICASSP-86, **3**:1593-1596, Tokyo, Japan, April 1986.

[20] P. Clarkson and R. Rosenfeld, *Statistical Language Modelling using CMU-Cambridge Toolkit,* Proc. ESCA EuroSpeech'97, 2707-2710, Rhodes, Greece, September 1997.

[21] S. Davis and P. Mermelstein, *Comparison of Parametric Representations of Monosyllabic Word Recognition in Continuously Spoken Sentences,* IEEE Trans. Acoustics, Speech, & Signal Processing, **28**(4):357-366, 1980.

[22] Dempster, A.P., M.M. Laird and D.B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society Series B (methodological), **39**:1-38, 1977.

[23] N. Deshmukh, A. Ganapathiraju, R.J. Duncan and J. Picone, *Human Speech Recognition Performance on the 1995 CSR Hub-3 Corpus* Proc. ARPA Speech Recognition Workshop, 129-134, Harriman, NY, February 1996.

[24] V. Digalakis and H. Murveit, *Genones: Optimization the Degree of Tying in a Large Vocabulary HMM-based Speech Recognizer,* Proc. IEEE ICASSP-94, **1**:537-540, Adelaide, Australia, April 1994.

[25] V. Digalakis, D. Rtichev and L.G. Neumeyer, *Speaker adaptation using constrained estimation of Gaussian mixtures*, IEEE Trans. on Speech & Audio, **3**(5):357-366, September 1995.

[26] J. Dreyfus-Graf, *Sonograph and Sound Mechanics,* J. Acoust. Soc. America, **22**:731, 1949.

[27] H. Dudley and S. Balashek, *Automatic Recognition of Phonetic Patterns in Speech,* J. Acoust. Soc. America, **30**:721, 1958.

[28] W.J. Ebel and J. Picone, *Human Speech Recognition Performance on the 1994 CSR Spoke 10 Corpus*, Proc. ARPA Spoken Language Systems Technology Workshop, 53-59, Austin, TX, January 1995.

[29] S. Furui, *Comparison of speaker recognition methods using statistical features and dynamic features,* IEEE Trans. on Acoustics, Speech & Signal Processing, **ASSP-29**, 342-350, 1981.

[30] M.J.F. Gales and S.J. Young, *An improved approach to hidden Markov model decomposition of speech and noise,* Proc. IEEE ICASSP-92, 233-236, San Francisco, CA, March 1992.

[31] M.J.F. Gales and S.J. Young, *Robust Continuous Speech Recognition using Parallel Model Combination,* Computer Speech & Language, **9**(4):289-307, October 1995.

[32] M.J.F. Gales, *Cluster Adaptive Training for Speech Recognition,* Proc. ICSLP'98, 1783-1786, Sydney, Australia, November 1998.

[33] M.J.F. Gales, *Semi-Tied Covariance Matrices for Hidden Markov Models,* IEEE Trans. on Speech and Audio, **7**(3):273-281, May 1999.

[34] J.L. Gauvain, G. Adda, L. Lamel and M. Adda-Decker, *Transcribing Broadcast News: The LIMSI Nov96 Hub4 System,* Proc. ARPA Speech Recognition Workshop, 56-63, Chantilly, VA, February 1997.

[35] J.L. Gauvain, S. Bennacef, L. Devillers, L. Lamel and R. Rosset, *Spoken Language component of the MASK Kiosk* in K. Varghese, S. Pfleger(Eds.) Human Comfort and security of information systems, Springer-Verlag, 1997. Also in

Proc. Human Comfort and Security Workshop, Brussels, Belguim, October 1995.

[36] J.L. Gauvain, J.J. Gangolf, and L. Lamel, *Speech Recognition for an Information Kiosk,* Proc. ICSLP'96, 849–852, Philadelphia, PA, October 1996.

[37] J.L. Gauvain, L. Lamel and G. Adda, *Partitioning and Transcription of Broadcast News Data,* Proc. ICSLP'98, **5**:1335-1338, Sydney, Australia, December 1998.

[38] J.L. Gauvain, L.F. Lamel and M. Adda-Decker, *Developments in Continuous Speech Dictation using the ARPA WSJ Task,* Proc. IEEE ICASSP-95, 65-68, Detroit, MI, May 1995.

[39] J.L. Gauvain and C.H. Lee, *Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains,* IEEE Trans. Speech & Audio Processing, **2**(2):291-298, April 1994.

[40] J.L. Gauvain, L. Lamel and G. Adda, *The LIMSI Broadcast News Transcription System*, Speech Communication, **37**(1-2):89-108, May 2002.

[41] L. Gillick and R. Roth, *A Rapid Match Algorithm for Continuous Speech Recognition,* Proc. DARPA Speech & Natural Language Workshop, 170-172, Hidden Valley, PA, June 1990.

[42] L. Gillick, Y. Ito and J. Young, *A Probabilistic Approach to Confidence Measure Estimation and Evaluation*, Proc. IEEE ICASSP-97, 879-882, Munich, Germany, April 1997.

[43] J.R. Glass, T.J. Hazen and I. L. Hetherington, *Real-time Telephone-based Speech Recognition in the Jupiter Domain,* Proc. IEEE ICASSP-99, **1**:61-64, Phoenix, AZ, March 1999.

[44] J. Godfrey, E. Holliman and J. McDaniel, *SWITCHBOARD: Telephone Speech Corpus for Research and Development,* Proc. IEEE ICASSP-92, 517-520, San Francisco, CA, March 1992.

[45] I.J. Good, *The Population Frequencies of Species and the Estimation of Population Parameters*, Biomterika, **40**(3/4):237-264, 1953.

[46] P.S. Gopalakrishnan, L.R. Bahl and R.L. Mercer, *A tree search strategy for large-vocabulary continuous speech recognition,* Proc. IEEE ICASSP-95, **1**:572-575, Detroit, MI, May 1995.

[47] R. Haeb-Umbach and H. Ney, *Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition,* Proc. ICASSP-92, **1**:13-16, March 1992.

[48] T. Hain, S.E. Johnson, A. Tuerk, P.C. Woodland and S.J. Young, *Segment Generation and Clustering in the HTK Broadcast News Transcription System,* Proc. DARPA Broadcast News Transcription & Understanding Workshop, 133-137, Landsdowne, VA, February 1998.

[49] A.G. Hauptmann, M. Witbrock and M. Christel, *News-on-Demand-'An Application of Informedia Technology',* Digital Libraries Magazine, September 1995.

[50] C.T. Hemphill, J.J. Godfrey, and G.R. Doddington, *The ATIS Spoken Language Systems Pilot Corpus,* Proc. DARPA Speech & Natural Language Workshop, Pittsburgh, PA, June 1990.

[51] H. Hermansky, *Perceptual linear predictive (PLP) analysis of speech,* J. Acoust. Soc. America, **87**(4):1738-1752, 1990.

[52] M.M. Hochberg, S.J. Renals, A.J. Robinson and D. Kershaw, *Large vocabulary continuous speech recognition using a hybrid connectionist-HMM system,* Proc. ICSLP'94, 1499-1502, Yokohama, Japan, September 1994.

[53] M.J. Hunt, *Signal Representation*, Chapter 1.3 of the State of the Art in Human Language Technology, (Cole et al, eds.), 1996. (http://www.cse.ogi.edu/CSLU/HLTsurvey/ch1node2.html)

[54] M. Hwang and X. Huang, *Subphonetic Modeling with Markov States - Senone,* Proc. IEEE ICASSP-92, **1**:33-36, San Francisco, CA, March 1992.

[55] M.Y. Hwang, X. Huang and F. Alleva, *Predicting Unseen Triphones with Senones,* Proc. IEEE ICASSP-93, **II**:311-314, Minneapolis, MN, April 1993.

[56] F. Jelinek, *Continuous Speech Recognition by Statistical Methods,* Proc. of the IEEE, **64**(4): 532-556, April 1976.

[57] F. Jelinek, *Statistical Methods for Speech Recognition,* Cambridge: MIT Press, 1997.

[58] F. Jelinek, B. Merialdo, S. Roukos and M. Strauss, *A Dynamic Language Model for Speech Recognition,* Proc. DARPA Speech & Natural Language Workshop, 293-295, Pacific Grove, CA, February 1991.

[59] F. deJong, J.L. Gauvain, J. deb Hartog and K. Netter, OLIVE*: Speech Based Video Retrieval,* Proc. CBMI'99, Toulouse, France, October 1999.

[60] Juang, B.-H., *Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains*, AT&T Technical Journal, **64**(6), 1985.

[61] S.M. Katz, *Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer,* IEEE Trans. Acoustics, Speech & Signal Processing, **ASSP-35**(3):400-401, March 1987.

[62] T. Kemp and A. Waibel, *Unsupervised Training of a Speech Recognizer: Recent Experiments,* Proc. ESCA Eurospeech'99, **6**:2725-2728, Budapest, Hungary, September 1999.

[63] D. Kershaw, A.J. Robinson and S.J. Renals, *The 1995 Abbot hybrid connectionist-HMM large-vocabulary recognition system,* Proc. ARPA Speech Recognition Workshop, 93-98, Harriman, NY, February 1996.

[64] R. Kneser and H. Ney, *Improved Clustering Techniques for Class-Based Statistical Language Modelling,* Proc. Eurospeech'93, 973-976, Berlin, September 1993.

[65] R. Kneser and H. Ney, *Improved backing-off for n-gram language modeling,* Proc. IEEE ICASSP-95, **1**:181-184, Detroit, MI, May 1995.

[66] F. Kubala, *Design of the 1994 CSR Benchmark Tests,* Proc. ARPA Spoken Language Systems Technology Workshop, 41-46, Austin, TX, January 1995.

[67] F. Kubala, T. Anastasakos, H. Jin, J. Makhoul, L. Nguyen, R. Schwartz and N. Yuan, *Toward Automatic Recognition of Broadcast News,* Proc. DARPA Speech Recognition Workshop, 55-60, Harriman, NY, February 1996.

[68] N. Kumar and A.G. Andreou, *Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition,* Speech Communication, **26**(4):283-297, December 1998.

[69] R. Kuhn, P. Nguyen, J.C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, and K. Field, M. Contolini, *Eigenvoices for Speaker Adaptation,* Proc. ICSLP'98, 1771-1774, Sydney, November 1998.

[70] L.F. Lamel and G. Adda, *On Designing Pronunciation Lexicons for Large Vocabulary, Continuous Speech Recognition,* Proc. ICSLP'96, **1**:6-9, Philadelphia, PA, October 1996.

[71] L.F. Lamel and R. DeMori, *Speech Recognition of European Languages,* Proc. IEEE Automatic Speech Recognition Workshop, 51-54, Snowbird, Utah, December 1995.

[72] L.F. Lamel and J.L. Gauvain, *Continuous Speech Recognition at LIMSI,* Proc. ARPA Workshop on Continuous Speech Recognition, 59-64, Stanford, CA, September 1992.

[73] L.F. Lamel and J.L. Gauvain, *A Phone-based Approach to Non-Linguistic Speech Feature Identification,* Computer Speech & Language, **9**(1):87-103, January 1995.

[74] L. Lamel, J.L. Gauvain, and G. Adda, *Lightly Supervised and Unsupervised Acoustic Model Training*, Computer, Speech & Language, **16**(1):115-229, January 2002.

[75] L.F. Lamel, S. Rosset, S.K. Bennacef, H. Bonneau-Maynard, L. Devillers and J.L. Gauvain, *Development of Spoken Language Corpora for Travel Information*, Proc. ESCA Eurospeech'95, **3**:1961-1964, Madrid, Spain, September 1995.

[76] K.-F. Lee, *Large-vocabulary speaker-independent continuous speech recognition: The SPHINX system,* PhD Thesis, Carnegie Mellon University, 1988.

[77] L. Lee and R.C. Rose, *Speaker Normalization Using Efficient Frequency Warping Procedures*, Proc. IEEE ICASSP-96, **1**:353-356, Atlanta, GA, May 1996.

[78] C.J. Leggetter and P.C. Woodland, *Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models,* Computer Speech & Language, **9**(2):171-185, April 1995.

[79] Liporace, L.R., *Maximum Likelihood Estimation for Multivariate Observations of Markov Sources*, IEEE Transactions on Information Theory, **IT-28**(5):729-734, 1982.

[80] R.P. Lippmann, *Speech recognition by machines and humans,* Speech Communication, **22**(1):1-15, July 1997.

[81] D. Liu and F. Kubala, *Fast Speaker Change Detection for Broadcast News Transcription and Indexing*, Proc. ESCA EuroSpeech'99, **3**:1031-1034, Budapest, Hungary, September 1999.

[82] Madcow, *Multi-site Data Collection for a Spoken Language Corpus,* Proc. DARPA Speech & Natural Language Workshop, 7-14, Harriman, NY, February 1992.

[83] L. Mangu, E. Brill, A. Stolcke, *Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks,* Computer, Speech and Language, 14(4):373-400, October 2000.

[84] B. Mak and E. Bocchieri, *Subspace distribution clustering for continuous observation density hidden Markov models,* Proc. Eurospeech'97, 107-110, Rhodes, Greece, September 1997.

[85] J.J. Mariani *Spoken Language Processing and Human-Machine Communication in the European Union Programs,* in G. Varile, ed., Eurospeech'97 EU Speech Projects Day report, Rhodes, Greece, September 1997.

[86] J.J. Mariani and L.F. Lamel, *An overview of EU programs related to conversational/interactive systems,* Proc. DARPA Broadcast News Transcription & Understanding Workshop, 247-253, Landsdowne, VA, February 1998.

[87] S. Martin, J. Liermann and H. Ney, *Algorithms for Bigram and Trigram Clustering,* Proc. Eurospeech'95, 1253-1256, Madrid, Spain, September 1995.

[88] M. Maybury (ed.), *News on Demand,* Special Section in the Communications of the ACM **43**(2), February 2000.

[89] D. Miller, R. Schwartz, R. Weischedel and R. Stone, *Named Entity Extraction from Broadcast News,* Proc. DARPA Broadcast News Workshop, 37-40, Herndon, VA, February 1999.

[90] M. Mohri, M. Riley, D. Hindle, A. Ljolie and F. Pereira, *Full Expansion of Context-Dependent Networks in Large Vocabulary Speech Recognition,* Proc. IEEE ICASSP-98, 665-668, Seattle, WA, May 1998.

[91] H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, *Large-Vocabulary Dictation using SRI's Decipher Speech Recognition System: Progressive*

Search Techniques, Proc. IEEE ICASSP-93, **II**:319-322, Minneapolis, MN, April 1993.

[92] H. Ney, *The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition,* IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-32**(2):263-271, April 1984.

[93] H. Ney, R. Haeb-Umbach, B.H. Tran and M. Oerder, *Improvements in Beam Search for 10000-Word Continuous Speech Recognition,* Proc. IEEE ICASSP-92, **I**:9-12, San Francisco, CA, March 1992.

[94] L. Nguyen and R. Schwartz, *Single-Tree Method for Grammar-Directed Search,* Proc. IEEE ICASSP-99, **2**:613-616, Phoenix, AZ, March 1999.

[95] J.J. Odell, *The Use of Decision Trees with Context Sensitive Phoneme Modelling,* MPhil Thesis, Cambridge University Engineering Dept, 1992.

[96] J.J. Odell, V. Valtchev, P.C. Woodland and S.J. Young, *A One Pass Decoder Design for Large Vocabulary Recognition,* Proc. ARPA Human Language Technology Workshop, 405-410, Princeton, NJ, March 1994.

[97] K. Ohtsuki, S. Furui, N. Sakurai, A. Iwasaki and Z.P. Zeang, *Recent Advances in Japanese Broadcast News Transcription,* Proc. ESCA Eurospeech'99, **2**:671-674, Budapest, Hungary, September 1999.

[98] P.A. Olsen and R.A. Gopinath, *Modeling Inverse Covariance Matrices by Basis Expansion,* Proc. IEEE ICASSP-02, 945-948, Orlando, FL, 2002.

[99] S. Ortmanns, H. Ney, and A. Eiden, *Language-model look-ahead for large vocabulary speech recognition,* Proc. ICSLP'96, 2095-2098, Philadelphia, PA, October 1996.

[100] S. Ortmanns, H. Ney, and X. Aubert, *A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition,* Computer, Speech and Language, **11**(1):43-72, January 1997.

[101] B.T. Oshika, V.W. Zue, R.V. Weeks, H. Neu and J. Aurbach, *The Role of Phonological Rules in Speech Understanding Research,* IEEE Trans. Acoustics, Speech, Signal Processing, **ASSP-23**, 104-112, 1975.

[102] M. Ostendorf, A. Kannan, O. Kimball and J.R. Rohlicek, *Continuous Word Recognition Based on the Stochastic Segment Model,* Proc. ARPA Workshop on Continuous Speech Recognition, 53-58, Stanford, CA, September 1992.

[103] D.S. Pallett, J.G. Fiscus, W.M. Fisher, J.S. Garofolo, B.A. Lund and M.A. Pryzbocki, *1993 Benchmark Tests for the ARPA Spoken Language Program,* Proc. ARPA Human Language Technology Workshop, 49-74, Princeton, NJ, March 1994.

[104] D.S. Pallett, J.G. Fiscus, W.M. Fisher, J.S. Garofolo, B.A. Lund, A.F. Martin and M.A. Przybocki, *1994 Benchmark Tests for the ARPA Spoken Language*

Program, Proc. ARPA Spoken Language Systems Technology Workshop, 5-36, Austin, TX, January 1995.

[105] D.S. Pallett, J.G. Fiscus, W.M. Fisher, J.S. Garofolo, A.F. Martin and M.A. Przybocki, *1995 Hub-3 Multiple Microphone Corpus Benchmark Tests,* Proc. ARPA Speech Recognition Workshop, 27-46, Harriman, NY, February 1996.

[106] D.S. Pallett, J.G. Fiscus, J.S. Garofolo, A.F. Martin and M.A. Przybocki, *1998 Broadcast News Benchmark Test Results: English and Non-English Word Error Rate Performance Measures,* Proc. DARPA Broadcast News Workshop, 5-12, Herndon, VA, February 1999.

[107] D.B. Paul, *An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model,* Proc. IEEE ICASSP-92, 405-409, San Francisco, CA, March 1992.

[108] D. Povey and P. Woodland, *Improved Discriminative Training Techniques For Large Vocabulary Continuous Speech Recognition*, Proc. IEEE ICASSP-01, Salt Lake City, May 2001.

[109] P. Price, *Evaluation of Spoken Language Systems: The ATIS Domain,* Proc. DARPA Speech and Natural Language Workshop, 91-95, Hidden Valley, PA, June, 1990.

[110] L.R. Rabiner and B.H. Juang, *An Introduction to Hidden Markov Models*, IEEE Acoustics Speech and Signal Processing Magazine, **ASSP-3**(1):4-16, January 1986.

[111] M.K. Ravishankar, *Efficient Algorithms for Speech Recognition,* PhD Thesis, Carnegie Mellon University, 1996.

[112] M.D. Riley, W. Byrne, M. Finke, S. Khudanpu, A. Ljojle, J. McDonough, H. Nock, M. Saraclar, C. Wooters and G. Zavaliagkos, *Stochastic pronunciation modelling from hand-labelled phonetic corpora,* Speech Communication, **29**(2-4):209-224, November 1999.

[113] R. Rosenfeld and X. Huang, *Improvements in Stochastic Language Modeling,* Proc. DARPA Workshop on Speech & Natural Language, 107-111, Harriman, NY, February 1992.

[114] R. Rosenfeld, *Adaptive Statistical Language Modeling,* Ph.D. Thesis, Carnegie Mellon University, 1994. (also Tech. rep. CMU-CS-94-138).

[115] R. Rosenfeld, *Two Decades of Statistical Language Modeling: Where Do We Go From Here?,* Proceedings of the IEEE, Special issue on Spoken Language Processing, **88**(8):1270-1278, August 2000.

[116] T. Schultza and A. Waibel, *Language-independent and langauge-adaptive acoustic modeling for speech recognition*, Speech Communication, **35**(1-2):31-51, August 2001.

[117] M. Schuster, *Memory-efficient LVCSR search using a one-pass stack decoder,* Computer Speech & Language, **14**(1):47-77, January 2000.

[118] R. Schwartz, S. Austin, F. Kubala and J. Makhoul, *New uses for N-Best Sentence Hypothesis, within the BYBLOS Speech Recognition System,* Proc. IEEE ICASSP-92, **I**:1-4, San Francisco, CA, March 1992.

[119] R. Schwartz, Y. Chow, S. Roucos, M. Krasner and J. Makhoul, *Improved Hidden Markov Modeling of Phonemes for Continuous Speech Recognition,* Proc. IEEE ICASSP-84, **3**:35.6.1-35.6.4, San Diego, CA, March 1984.

[120] S. Sekine and R. Grishman, *NYU Language Modeling Experiments for the 1995 CSR Evaluation,* Proc. ARPA Speech Recognition Workshop, 123-128, Harriman, NY, February 1996.

[121] B. Shahshahani, *A Markov Random Field Approach to Bayesian Speaker Adaptation,* Proc. IEEE ICASSP-95, 697-700, Detroit, MI, May 1995.

[122] R. Schwartz, H. Jin, F. Kubala and S. Matsoukas, *Modeling Those F-Conditions – Or Not,* Proc. DARPA Speech Recognition Workshop, 115-118, Chantilly, VA, February 1997.

[123] K. Seymore and R. Rosenfeld, *Scalable backoff language models*, Proc. ICSLP'96, **1**:232-235, Philadelphia, PA, October 1996.

[124] M. Siegler, U. Jain, B. Raj and R. Stern, *Automatic Segmentation, Classification and Clustering of Broadcast News Audio,* Proc. DARPA Speech Recognition Workshop, 97-99, Chantilly, VA, February 1997.

[125] M. Siu and H. Gish, *Evaluation of word confidence for speech recognition systems*, Computer Speech & Language, **13**(4):299-318, October 1999.

[126] A. Stolcke, *Entropy-based Pruning of Backoff Language Models*, Proc. DARPA Broadcast News Transcription & Understanding Workshop, 270-274, Landsdowne, VA, February 1998.

[127] S. Takahashi and S. Sagayama, *Four-level Tied Structure for Efficient Representation of Acoustic Modeling,* Proc. IEEE ICASSP-95, 520-523, Detroit, MI, May 1995.

[128] L.F. Uebel and P.C. Woodland, *An Investigation into Vocal Tract Length Normalization,* Proc. ESCA Eurospeech'99, 2527-2530, Budapest, Hungary, September 1999.

[129] D.A. van Leeuwen, L.G. van den Berg and H.J.M. Steeneken, *Human Benchmarks for Speaker Independent Large Vocabulary Recognition Performance,* Proc. ESCA Eurospeech'95, 1461-1464, Madrid, Spain, September 1995.

[130] T.K. Vintsyuk, *Speech discrimination by dynamic programming,* Kibnernetika, **4**:81, 1968.

[131] T.K. Vintsyuk, *Elements-wise recognition of continuous speech composed of words from a specified dictionary,* Cybernetics, **7**:133-143, March-April 1971.

[132] W. Wahlster, *Verbmobil: Translation of Face-to-Face Dialogs,* Proc. ESCA Eurospeech'93, Berlin, Germany, **Plenary**, 29-38, September 1993.

[133] A. Waibel, P. Geutner, L. Mayfield Tomokiyo, T. Schultz and M. Woszczyna, *Multilinguality in Speech and Spoken Language Systems*, Proceedings of the IEEE, Special Issue on Spoken Language Processing, **88**(8):1297-1313, August 2000.

[134] F. Walls, H. Jin, S. Sista and R. Schwartz, *Probabilistic Models for Topic Detection and Tracking,* Proc. IEEE ICASSP-99, **1**:521-524, Phoenix, AZ, March 1999.

[135] S. Wegmann, F. Scattone, I. Carp, L. Gillick, R.Roth and J. Yamron, *Dragon Systems' 1997 Broadcast News Transcription System,* Proc. DARPA Broadcast News Transcription & Understanding Workshop, 60-65, Landsdowne, VA, February 1998.

[136] S. Wegmann, P. Zhan, and L. Gillick, *Progress in Broadcast News Transcription at Dragon Systems,* Proc. IEEE ICASSP-99, 33-36, Phoenix, AZ, March 1999.

[137] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig and A. Stolcke, *Neural-Network based Measures of Confidence for Word Recognition,* Proc. IEEE ICASSP-97, 887-890, Munich, Germany, April 1997.

[138] F. Wessel, K. Macherey and R. Schlüter, *Using word probabilities as confidence measures,* Proc. IEEE ICASSP-98, 225-228, Seattle, WA, May 1998.

[139] F. Wessel and H. Ney, *Unsupervised training of acoustic models for large vocabulary continuous speech recognition*, Proc. IEEE ASRU'01, Madonna di Campiglio, Italy, December 2001.

[140] I.H. Witten and T.C. Bell, *The Zero Frequency problem: Estimating the problems of Novel Events in Adaptive tex Compression*, Proc. IEEE Trans. on Information Theory, **37**():1085-1094, July 1991.

[141] P.C. Woodland and D. Povey, *Large scale discriminative training of hidden Markov models for speech recognition,* Computer, Speech and Language, **16**(1):25-47, January 2002.

[142] P.C. Woodland, C.J. Leggetter, J.J. Odell, V. Valtchev and S.J. Young, *The development of the 1994 HTK large vocabulary speech recognition system,* Proc. ARPA Spoken Language Systems Technology Workshop, 104-109, Austin, TX, January 1995.

[143] P.C. Woodland, M.J.F. Gales, D. Pye and V. Valtchev, *The HTK large vocabulary recognition system for the 1995 ARPA H3 task,* Proc. ARPA Speech Recognition Workshop, 99-104, Harriman, NY, February 1996.

[144] J.P. Yamron, I. Carp, L. Gillick, S. Lowe and P. van Mulbregt, *A Hidden Markov Approach to Text Segmentation and Event Tracking*, Proc. IEEE ICASSP-98, **1**:333-336, Seattle, WA, May 1998.

[145] S.J. Young, *A Review of Large-Vocabulary Continuous Speech Recognition,* IEEE Signal Processing Magazine, **13**(5):45-57, September 1996.

[146] S.J. Young, M. Adda-Decker, X. Aubert, C. Dugast, J.L. Gauvain, D.J. Kershaw, L. Lamel, D.A. Leeuwen, D. Pye, H.J.M. Steeneken, A.J. Robinson and P.C. Woodland, *Multilingual large vocabulary speech recognition: the European SQALE project,* Computer Speech & Language, **11**(1):73-89, January 1997.

[147] S.J. Young and L. Chase, *Speech recognition evaluation: a review of the U.S. CSR and LVCSR programmes,* Computer Speech & Language, **12**(4):263-279, October 1998.

[148] S.J. Young, J.J. Odell and P.C. Woodland, *Tree-Based State Tying for High Accuracy Acoustic Modeling,* Proc. ARPA Human Language Technology Workshop, 307-312, Princeton, NJ, March 1994.

[149] S.J. Young and P.C. Woodland, *The Use of State Tying in Continuous Speech Recognition,* Proc. ESCA Eurospeech'93, **3**:2203-2206, Berlin, Germany, September 1993.

[150] G. Zavaliagkos and T. Colthurst, *Utilizing Untranscribed Training Data to Improve Performance*, DARPA Broadcast News Transcription and Understanding Workshop, Landsdowne, VA, 301-305, February 1998.

[151] G. Zavaliagkos, R. Schwartz and J. McDonough, *Maximum a Posteriori Adaptation for Large Scale HMM Recognizers,* Proc. IEEE ICASSP-95, 725-728, Detroit, MI, May 1995.

[152] V. Zue, J. Glass, M. Phillips and S. Seneff, *The MIT* SUMMIT *Speech Recognition System: A Progress Report*, Proc. DARPA Speech & Natural Language Workshop, 179-189, Philadelphia, PA, February 1989.

6

Toward Spontaneous Speech Recognition and Understanding

Sadaoki Furui
Tokyo Institute of Technology

CONTENTS

6.1 Introduction

Speech recognition systems are expected to play important roles in an advanced multi-media society with user-friendly human-machine interfaces [1]. The field of automatic speech recognition has witnessed a number of significant advances in the past 5-10 years, spurred on by advances in signal processing, algorithms, computational architectures, and hardware. These advances include the widespread adoption of a statistical pattern recognition paradigm, a data-driven approach which makes use of a rich set of speech utterances from a large population of speakers, the use of stochastic acoustic and language modeling, and the use of dynamic programming-based search methods [2, 3, 4].

The state-of-the-art in automatic speech recognition can be addressed in several ways. Figure 6.1 illustrates the progress of speech recognition and understanding technology according to generic application areas, ranging from isolated word or command recognition to natural conversation between human and machine. The complexity of these generic application areas is characterized along two dimensions: the size of the vocabulary and the speaking style. It should be obvious that the larger the vocabulary, the more difficult the application task. Similarly, the degree of constraints in the speaking style has a very direct influence on the complexity of the

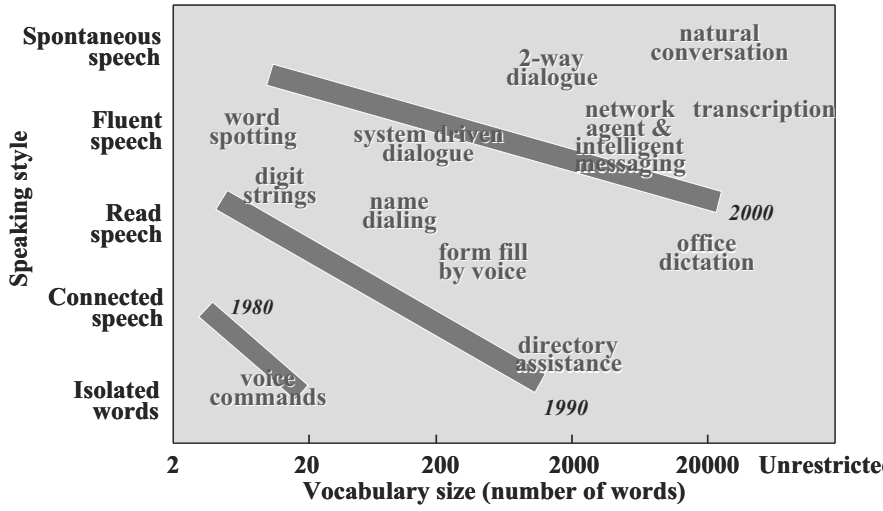

FIGURE 6.1
Progress of spoken language technology along the dimensions of vocabulary size and speaking styles.

application; a free conversation full of slurring and extraneous sounds such as "uh", "um", and partial words is far more difficult than words spoken in a rigidly discrete manner. Thus, the difficulty of an application grows from the lower left corner to the upper right corner in the figure. The three bars in the figure demarcate the applications that can and cannot be supported by the technology for viable deployment in the corresponding time frame. It should be noted that these three bars are not parallel; which means that the progress of spontaneous speech recognition and understanding is much slower than that of more rigidly spoken utterances.

Common features of state-of-the-art speech recognition systems exist in using cepstral parameters and their regression coefficients as speech features, triphone HMMs as acoustic models, vocabularies of several thousand or several ten thousand entries, and statistical language models such as bigrams and trigrams. Such methods have been applied not only to English but also to French, German, Italian, and Japanese; and, although there are several language-specific characteristics, similar recognition results have been obtained. Recently, tasks using natural conversational speech have been actively investigated. In spite of the remarkable recent progress, we are still far behind our ultimate goal of understanding free spontaneous speech uttered by any speaker in any environment.

Read speech and similar types of speech, e.g. that from reading newspapers or from news broadcast, can be recognized with accuracy higher than 90% using the state-of-the-art speech recognition technology. However, recognition accuracy drastically decreases for spontaneous speech. This decrease is due to the fact that the acoustic and

linguistic models used have generally been built using written language or speech from written language. Unfortunately spontaneous speech and speech from written language are very different both acoustically and linguistically. Broadening the application of speech recognition thus crucially depends on raising the recognition performance for spontaneous speech. In order to increase the recognition performance for spontaneous speech, it is crucial to build acoustic and language models for spontaneous speech. Methods applying statistical language modeling such as bigrams and trigrams of words or morphemes to spontaneous speech corpus may not be adequate. Our knowledge of the structure of spontaneous speech is currently inadequate to achieve the necessary breakthroughs. Although spontaneous speech effects are quite common in human communication and may be expected to increase in human machine discourse as people become more comfortable conversing with machines, modeling of speech disfluencies is only just beginning. Recognition of spontaneous speech will require a paradigm shift from speech recognition to understanding where underlying messages of the speaker are extracted, instead of transcribing all the spoken words [5].

Much of our thinking about speech recognition has been focused on its use as an interface in human-machine interactions mostly for information access and extraction. With increases in cellular phone use and dependence on networked information resources, and as rapid access to information becomes an increasingly important economic factor, telephone access to data and telephone transactions will no doubt rise dramatically. There is a growing interest, however, in viewing speech not just as a means to access information, but as, itself, a source of information. Important attributes that would make speech more useful in this respect include: random access, sorting (e.g., by speaker, by topic, by urgency), scanning, and editing. How could our lives be changed by such tools? Enabling such a vision challenges our systems still further in noise robustness and in spontaneous speech effects.

We can envision a great information revolution on par with the development of writing systems, if we can successfully meet the challenges of speech both as a medium for information access and as itself a source of information. Speech is still the means of communication used first and foremost by humans, and only a small percentage of human communication is written. Automatic speech understanding can add many of the advantages normally associated only with text (random access, sorting, and access at different times and places) to the many benefits of speech. Making this vision a reality will require significant advances.

6.2 Four Categories of Speech Recognition Tasks

Speech recognition tasks can be classified into four categories as shown in Table 6.1 according to two criteria: whether it is targeting utterances from human to human or human to computer, and whether the utterances have a dialogue or monologue style.

TABLE 6.1
Categorization of speech recognition tasks

	Dialogue	Monologue
Human to human	(Category I) Switchboard, Call Home (Hub 5), meeting task	(Category II) Broadcasts news (Hub 4), lecture, presentation, voice mail
Human to machine	(Category III) ATIS, Communicator, information retrieval, reservation	(Category IV) Dictation

The table lists typical tasks for each category.

Most of the practical application systems widely used now are classified as Category III, recognizing the utterances in human-computer dialogues such as in the airline information services task. DARPA-sponsored projects including ATIS and Communicator are laying the foundations of these systems. Unlike other categories, the systems in the Category III are usually designed and developed after clearly defining the application/task. The machine that we have attempted to design so far is, almost without exception, limited to the simple task of converting a speech signal into a word sequence and then determining, from the word sequence, the meaning that is "understandable". Here, the set of understandable messages is finite in number, each being associated with a particular action (e.g., route a call to a proper destination or issue a buy order for a particular stock). In this limited sense of speech communication, the focus is detection and recognition rather than inference and generation.

Category I targets human-to-human dialogues and includes DARPA-sponsored Switchboard and Call Home (Hub 5) tasks. Speech recognition research with the aim of making minutes of meetings has recently started in this category.

One of the typical tasks belonging to the Category IV, that targets the recognition of monologues performed when people are talking to a computer, is dictation. Various commercial software for such purposes has been developed.

Tasks belonging to the Category II, that target recognizing human-to-human monologues, include transcription of broadcast news (Hub 4), lectures, presentations, and voice mails. Speech recognition research in this category has recently become very active.

Various research has made clear that the utterances spoken by people talking to computers, such as those in the Categories III and IV, especially when the speaker is conscious, are acoustically as well as linguistically very different from those spoken to other people, such as those in Categories I and II. Even in utterances spoken to people, the acoustic and linguistic characteristics of monologues, such as lectures, presentations, and voice mails, are largely different from that of daily dialogues.

Since the utterances in the Category II are made with the expectation that the audience can correctly understand what is spoken in the one-way communication, they are relatively easier to perform recognition on than the utterances in Category I.

If high recognition performance is achieved, a wide range of applications, such as making lecture notes, records of presentations and closed captions, archiving these records, their retrieval, and the retrieval of voice mails, will be realized.
Since the utterances in the Category IV are made with the expectation that his/her utterances are exactly converted into texts with correct characters, their spontaneity is much lower than that in the Category III. In the four categories, spontaneity is considered to be the highest in Category I and the lowest in Category IV.
Among these four categories, this chapter first briefly reviews Categories I, II, and III, and then it focuses on Category II. A large-scale national project to investigate the issues of spontaneous speech recognition is introduced.

6.3 Spontaneous Speech Recognition and Understanding - Review

6.3.1 Category I (human-to-human dialogue)

Switchboard [6] is a DARPA-sponsored large multispeaker corpus of spontaneous conversational telephone bandwidth speech and text for research on large vocabulary speech recognition and speaker authentication. About 2500 conversations by 500 speakers from around the U.S. were collected automatically over T1 lines. In each conversation, two speakers were asked to discuss one of 70 different topics such as pets, crime, or air pollution. These conversations are of duration three to ten minutes, five minutes in average, and spoken by paid volunteers of both sexes in every major dialect of American English. This amounts to over 250 hours of speech and nearly three million words of text. Recognition of utterances in the Switchboard corpus is a very challenging task.
Oral communication is transient but many important decisions, social contracts, and fact findings are first carried out orally, documented in written form, and later retrieved. Humans spend a lot of time transforming oral communications into written documents. Research focusing on automatic meeting record creation and access has been conducted [7]. The research aims at a realistic meeting scenario, the corresponding speech recognition problems, the analysis of retrieval performance, the generation of readable summaries and a practical user interface. Meeting recognition is a very challenging LVCSR task parallel to that of Hub 5 (Switchboard) and Hub 4 (Broadcast News). The difficulty is due to three reasons. First, the conversational style - meetings consists of uninterrupted continuous recordings with multiple speakers talking in a conversational style. Second, the lack of training data - meeting data is highly specialized depending on the topic and participants, therefore large databases cannot be provided on demand. As a consequence, the research has focused on the question of how to build LVCSR systems for new tasks and languages using limited amounts of training data. Third, the degraded recording conditions: to minimize interference a clip-on lapel microphone was chosen instead of a close-

talking headset. This comes at the cost of significant channel cross-talk.

6.3.2 Category II (human-to-human monologue)

The DARPA-sponsored Hub 4 project has been a driving force behind research on human-to-human monologue speech recognition since 1995 [8]. In this project, television and radio news broadcasts are recorded and annotated. The materials consist of what has been termed "found speech" - "found" in news broadcasts, in contrast with the specially recorded "read" speech investigated in the former DARPA "North America Business (NAB) news" project. It proved to offer a rich assortment of technical challenges to the community, including varied speaking styles, foreign-accented English, the presence of background music and both full, as well as reduced-bandwidth channel effects. The lowest word error rates in the 1998 benchmark test results for the low-noise baseline, F0, and spontaneous, F1, conditions were 7.8% and 14.4%, respectively. The database was later extended to include Mandarin Chinese and Spanish.

In 1998, a new task ("spoke") was added to Hub 4 to examine the effectiveness of broadcast news recognition technology in generating information rich entities and to begin to move the research focus from simple transcription toward spoken information understanding. The task involved the recognition and identification of the following types of information entities in the broadcast news stream: named entities (person, location, and organization), temporal expressions (date and time), and numeric expressions (monetary and percentage).

In Japan, Japanese broadcast-news speech transcription systems have been researched and developed by NHK broadcasting company R&D lab and by several universities [9, 10]. The language models were constructed using broadcast-news manuscripts taken from NHK TV news broadcasts. Since Japanese sentences are written without spaces between words and there is no clear definition of words, the broadcast-news manuscripts were segmented into words (morphemes) using a morphological analyzer to calculate word n-gram language models. Many Japanese words have multiple readings and the correct one can only be decided according to the context. Therefore, language models in which a word with multiple readings is split into different language model entries according to those readings have been constructed. Since various characters, such as Chinese characters, multiple types of Japanese characters, numbers, and alphabets, are used in Japanese text, it is hard to type Japanese text in real time. Therefore, computer-based systems are indispensable for online Japanese closed captioning. NHK started the closed captioning using a real-time speech recognizer followed by manual correction of recognition errors in March 2000. Since recognition accuracy for spontaneous speech is not yet satisfactory, closed captioning is provided only for the speech uttered by anchors in the studio.

With the increasing number of different media sources for information dissemination, there is a rapidly growing need for fast automatic processing of audio data stream. Automation of audio segmentation, transcription, and indexation is indispensable. A spoken document indexing and retrieval system combining a state-of-

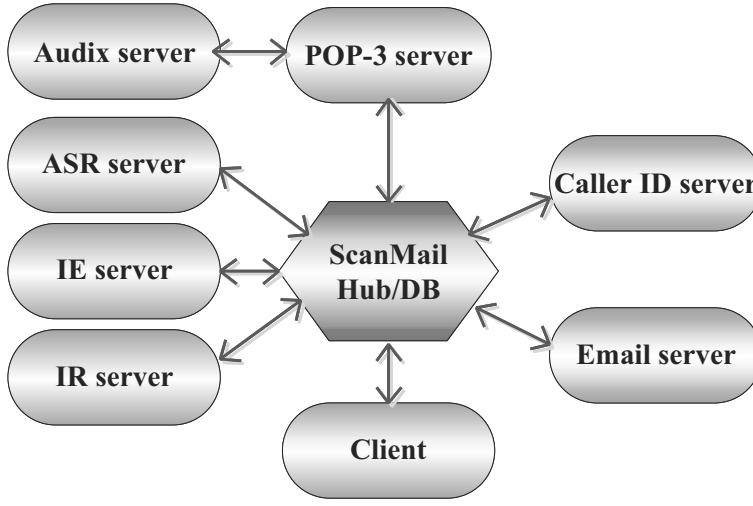

FIGURE 6.2
The SCANMail architecture [12].

the-art speech recognizer with a text-based information retrieval (IR) system has been investigated [11]. With query expansion using commercial transcripts, comparable mean precisions have been obtained on manual reference transcriptions and automatic transcriptions with a word error rate of 21.5%.

Voicemail speech recognition presents a challenging problem, since it is characterized by a variety of speaking rates, accents, tasks, and acoustic conditions. Additionally, phenomena such as disfluencies, restarts, repetitions, and broken words are common. In contrast to natural dialogue, voicemail speech is monologue, that is a "one-way" communication: speakers do not receive any direct feedback when they leave messages. The telephone channel also poses problems of low bandwidth and signal to noise ratio, since there are no restrictions on the location or type of telephone used to leave a voicemail message.

SCANMail [12] is a system that employs automatic speech recognition (ASR), IR, information extraction (IE), and human computer interaction (HCI) technology to permit users to browse and search their voicemail messages by content through a graphical user interface (GUI). The SCANMail client also provides note-taking capabilities as well as browsing and querying features. An email server sends the original message plus its transcription to mailing address specified in the user's profile. Figure 6.2 shows the architecture of the system. The language model for ASR is a Katz-style backoff trigram trained on 700,000 words from the transcriptions of the 60 hour training set.

An important issue related to the development of integrated voice/data communications is that of speech summarization: given a spoken passage, produce a short textual precis of its content. A system that transmits text summaries of a user's in-

coming voicemail messages, using the GSM short message service (SMS), reducing the need for users to listen to all of their messages, has been investigated [13]. Voice mail summarization differs from text summarization or abstracting, since it does not assume perfect transcriptions and is concerned with summarizing brief spoken messages (average duration about 40s) into terse (140 character) SMS summaries. The system uses a data-driven approach to summarizing spoken audio transcripts utilizing lexical and prosodic features. The approach has been evaluated on the IBM Voicemail corpus, demonstrating that it is possible and desirable to avoid complete commitment to a single best classifier or feature set.

A Japanese national project on spontaneous speech corpus and processing technology was initiated in 1999. This project aims to build a large-scale monologue spontaneous speech corpus and create spontaneous speech recognition and summarization technology. Details will be explained in Section 6.4.

6.3.3 Category III (human-to-machine dialogue)

There is a growing interest in mobile communication systems that allow users to use their voices to do more than speaking to other people; examples include accessing information services and interaction with booking services.

Providing voice interaction capability as a part of multimedia user experience is believed to add naturalness and efficiency to human-computer interactions. Numerous commercial spoken dialog systems are currently being deployed, primarily for access to information over the telephone. There are, however, major open research issues that challenge the deployment of completely natural and unconstrained voice interactions even for limited task domains. These primarily arise because the state-of-the-art in automatic speech recognition and understanding is far from perfect.

One of the simple speech understanding tasks that has been attempted is DARPA's Air Travel Information System (ATIS). In this task, the user talks to the machine to obtain flight information using natural speech such as "I would like to leave San Francisco for New York on December first, please list the available flights"; "How much does the flight cost from Denver to Dallas?"

The DARPA Communicator project is a multi-year multi-site project launched in 1999 [14]. The aim of the project is to construct a computer system that plays the role of a travel agent speaking by telephone with a customer. Ideally, this system will perform just as a human would: conversing with the user to determine the outline of the desired itinerary, querying airline databases to establish flight availability, reporting suitable flights to the user, answering questions to resolve uncertainties or misunderstandings, and finally booking the trip.

The DARPA Communicator dialog architecture is hub centric as shown in Figure 6.3 [15]. The hub is a programmable traffic router that is responsible for invoking the different servers in the system and routing messages between them. The hub architecture does not define the functionality but instead provides standard APIs. Therefore the servers depicted in the figure represent a particular instantiation of the Communicator architecture. The servers operate through callback functions that are invoked by the hub. The hub itself is event driven: upon receiving a new frame

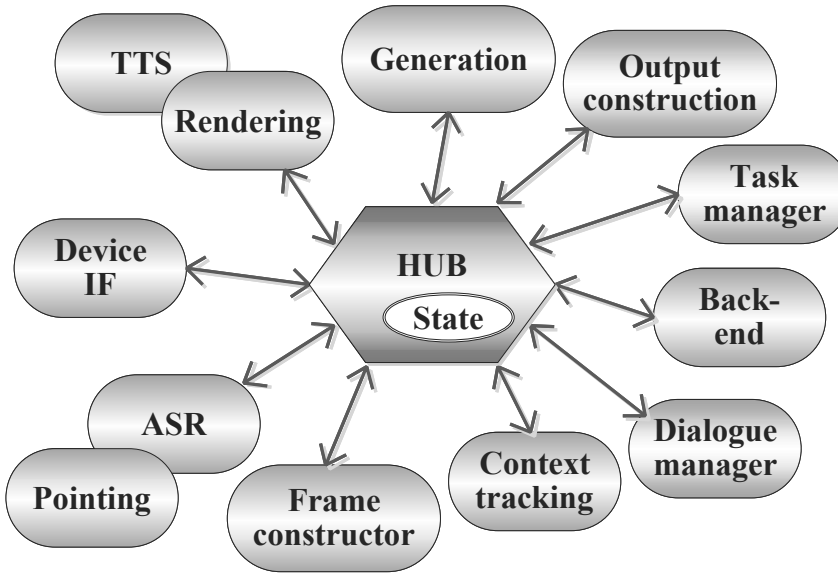

FIGURE 6.3
AT&T Communicator architecture [15].

message, it finds and invokes the appropriate callback functions and passes the frame to the destination servers.

The AT&T Communicator system is focusing on issues related to the design of mixed-initiative systems. The idea of mixed-initiative systems is to combine the flexibility of a user-initiative system with the constrained problem-solving nature of a system-initiative system. For example, a reasonable response to the query "Show me the flights" could be "Please tell me where you would like to fly." Given the state-of-the-art in ASR technology, mixed-initiative system design needs to trade-off between the degree of initiative allowed and the ASR performance.

For the call routing type of application, the problem is essentially that of pattern recognition. The observation is the query sentence, which contains a sequence of words. The classes for recognition are the actions (e.g., routing the call to a proper department). There can be several layers of approaches to this problem, depending on the depth of the linguistic inference that the system is designed to pursue. The simplest approach is to assume that in most query sentences, the intended action is going to be expressed in specific terms, spoken isolation or possibly embedded in a natural utterance. With the assumption that actions are likely to be expressed in keywords, the system can just employ keyword-spotting techniques to perform the task. This kind of system is simple to implement.

Another more complex approach that has been attempted takes into account all the words in the utterance, but without paying particular attention to the sequential order of the words. The method of information network [16] or latent semantic analysis [17] has been proposed with reasonable success. These methods use a correlation matrix or network between the actions and the occurrence of words to facilitate the decision process. Compared to keyword-spotting, these methods do not separate a priori words that are keywords and those that are not. They implicitly associate a (continuously valued) significance level between the appearance of a word and the intended action.

6.4 Japanese National Project on Spontaneous Speech Corpus and Processing Technology

6.4.1 Project Overview

For building language models for spontaneous speech, large spontaneous speech corpora are indispensable. In this context, a Science and Technology Agency Priority Program entitled "Spontaneous Speech: Corpus and Processing Technology" started in Japan in 1999 [18]. The project will be conducted over a five-year period under the following three major themes as shown in Figure 6.4.

1. Building a large-scale spontaneous speech corpus, Corpus of Standard Japanese (CSJ), consisting of roughly 7M words with the total speech length of 700 hours. Mainly recorded will be monologues such as lectures, presentations and news commentaries. The recordings will be manually given orthographic and phonetic transcription. One-tenth of the utterances, hereafter referred to as the core, will be tagged manually and used for training a morphological analysis and part-of-speech (POS) tagging program for automatically analyzing all of the 700-hour utterances. The core will also be tagged with para-linguistic information including intonation.

2. Acoustic and linguistic modeling for spontaneous speech understanding using linguistic as well as para-linguistic information in speech.

3. Investigating spontaneous speech summarization technology.

The technology created in this project is expected to be applicable to wide areas such as indexing of speech data (broadcast news, etc.) for information extraction and retrieval, transcription of lectures, preparing minutes of meetings, closed captioning, and aids for the handicapped.

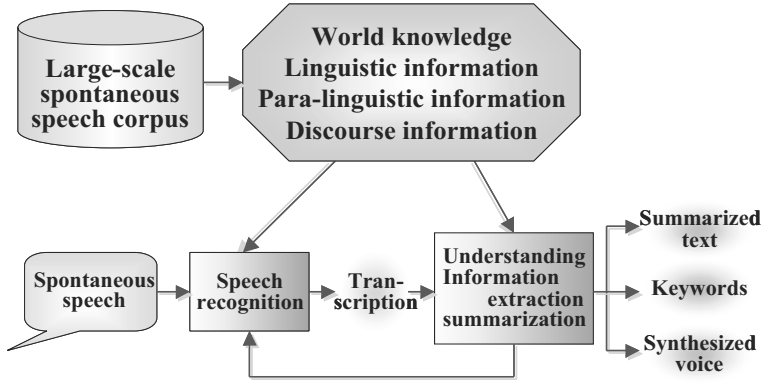

FIGURE 6.4
Overview of the Japanese national project on spontaneous speech corpus and processing technology.

6.4.2 Corpus

CSJ is the corpus of spontaneous monologue of standard Japanese [19]. More precisely, CSJ contains speech uttered to multiple listeners in a more or less formal social setting. The two main sources of spontaneous monologue for CSJ are live recording of various academic conferences/meetings, referred to as Academic Presentation (AP) hereafter, such as the Acoustical Society of Japan (ASJ) meetings, and studio recording of informal free public speech made by paid voluntary subjects, Simulated Public Speech (SPS). The SPS includes a wide variety of topics including the subjects' experiences in their daily lives. The AP speech, which is expected to have logical and concise discourse structure, is the target of the spontaneous speech recognition and summarization system that is developed in the project.

SPS is added to AP for several reasons; the most important is the skewed distribution of the age and sex of AP speakers. Most AP speakers are male graduate students in their twenties or early thirties. This is especially true with engineering-oriented societies. SPS speakers were recruited so that they showed a balanced distribution both in sex and age, ranging from early twenties to sixties. Another reason of adding SPS is the lexical bias of the AP vocabulary. The vocabulary of AP is deeply biased by the existence of field-specific technical terms. Finally, SPS is expected to be more spontaneous than AP; this is crucial for the linguistic study of spontaneous speech.

Figure 6.5 shows the design of the CSJ in terms of its data size. The total size of CSJ is seven million words. This amount is supposed to be the minimum requisite for the construction of a workable language model for speech recognition. Digitized speech (16kHz, 16bit linear), detailed transcription, and POS annotation are to be provided for the total body of CSJ. POS tagging of the corpus beyond the core will be automated.

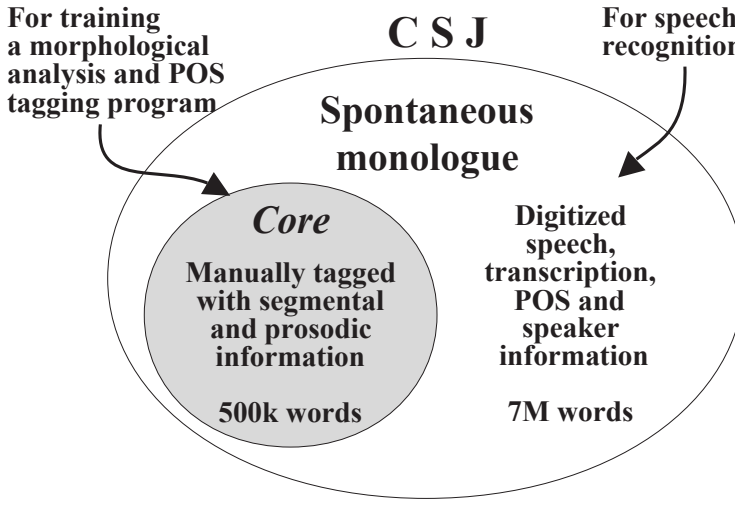

FIGURE 6.5
Overall design of the Corpus of Spontaneous Japanese.

6.5 Automatic Transcription of Spontaneous Presentation

6.5.1 Recognition Task

Using the CSJ corpus, preliminary recognition experiments are being conducted at Tokyo Institute of Technology as well as at several other universities participating in the project. In this experiment, presentation speech uttered by 10 male speakers was used as a test set of speech recognition [22]. Table 6.2 shows an outline of the test set.

6.5.2 Language and Acoustic Modeling

Sounds are digitized and segmented into utterances using silence periods longer than 500ms. Feature vectors have 25 elements consisting of 12 MFCC, their delta, and the delta log energy. Cepstral mean subtraction (CMS) is applied to each utterance. The following two corpora are used for training the language and acoustic models.

CSJ: A part of the corpus completed by the end of December 2000, consisting of approximately 1.5M words of transcriptions, is used. The training set consists of 610 presentations; 274 AP and 336 SPS presentations.

Web corpus**:** Transcribed presentations consisting of approximately 76k sentences with 2M words have been collected from the World Wide Web. Spontaneous

TABLE 6.2
Recognition test set of presentations

ID	Conference name	Length[min]
A22	Acoust. Soc. Jap.	28
A23	Acoust. Soc. Jap.	30
A97	Acoust. Soc. Jap.	12
P25	Phonetics Soc. Jap.	27
J01	Soc. Jap. Linguistics	57
K05	National Lang. Res. Inst.	42
N07	Assoc. Natural Lang. Proc.	15
S05	Assoc. Socioling. Sciences	23
Y01	Spont. Speech Corpus Meeting	14
Y05	Spont. Speech Corpus Meeting	15

speech usually includes various filled pauses but they are not included in this presentation corpus. An effort is thus made to add filled pauses to the presentation corpus based on the statistical characteristics of the filled pauses. The topics of the presentations cover wide domains including social issues and memoirs.

The following two language models, denoted as **SpnL** and **WebL**, have been constructed. Each model consists of bigrams and reverse trigrams with backing-off. Their vocabulary sizes are 30k words.

SpnL: Made using the 610 presentations in the CSJ. The speakers have no overlap with those of the test set. Since there are no punctuation marks in the transcription, commas are inserted when a silence period of 200ms or longer is encountered.

WebL: Made using the text of our Web corpus.

The following two tied-state triphone HMMs have been made, both having 2k states and 16 Gaussian mixtures in each state.

SpnA: Using 338 presentations in the CSJ uttered by male speakers (approximately 59 hours). The speakers have no overlap with those in the test set.

RdA: Using approximately 40 hours of read speech uttered by many speakers.

6.5.3 Recognition Results

Figure 6.6 presents the test-set perplexity of trigrams and the out-of-vocabulary (OOV) rate for each presentation, comparing the two language models. The perplexity and OOV of **SpnL** made from the CSJ are clearly better than that of the web-based model, **WebL**. **WebL** shows high perplexity and OOV rate, since it was edited as a text and their topics are much more diversified than those of the test set.

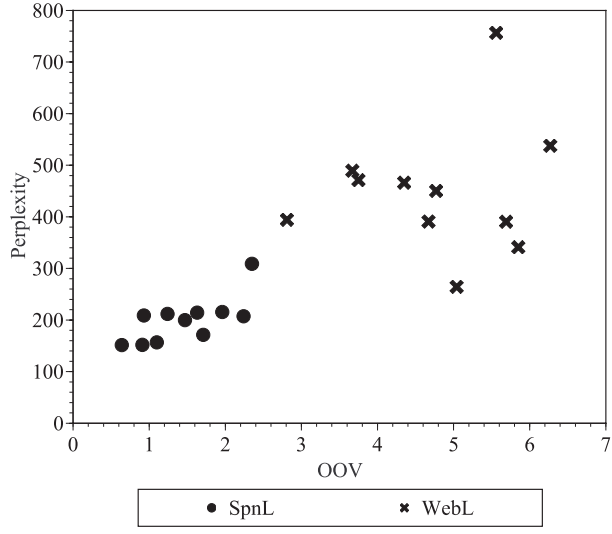

FIGURE 6.6
Test-set perplexity and OOV rate for the two language models.

Figure 6.7 shows recognition results for the combinations of the two language models, **SpnL** and **WebL**, and the two acoustic models, **SpnA** and **RdA**. Fillers are counted as words and included in calculating the accuracy. It is clearly shown that **SpnL** achieves much better results than **WebL**, and **SpnA** gives much better results than **RdA**. These results indicate that it is crucial to make language models from a spontaneous speech corpus to adequately recognize spontaneous speech. It is also suggested that acoustic models made from CSJ have better coverage of triphones and better matching of acoustic characteristics corresponding to the speaking style and also have better matching of recording conditions with the test set. The mean accuracy for the combination of **SpnL** and **SpnA** is 65.3%.

As shown in Figure 6.7, the word accuracy largely varies from speaker to speaker. There exist many factors that affect the accuracy of spontaneous speech recognition. They include individual voice characteristics, speaking manners, and noise like coughs. Although all utterances were recorded using the same close-talking microphones, acoustic conditions still varied according to the recording environment.

A batch-type unsupervised adaptation method has been incorporated to cope with the speech variation due to speakers and recording environment. The MLLR method using a binary regression class tree to transform Gaussian mean vectors is employed [20]. The regression class tree is made using a centroid-splitting algorithm. The actual classes used for transformation are determined on run time according to the amount of data assigned to each class [21].

The adaptation is performed based on recognition results, and no confidence measure is applied. The following steps are performed:

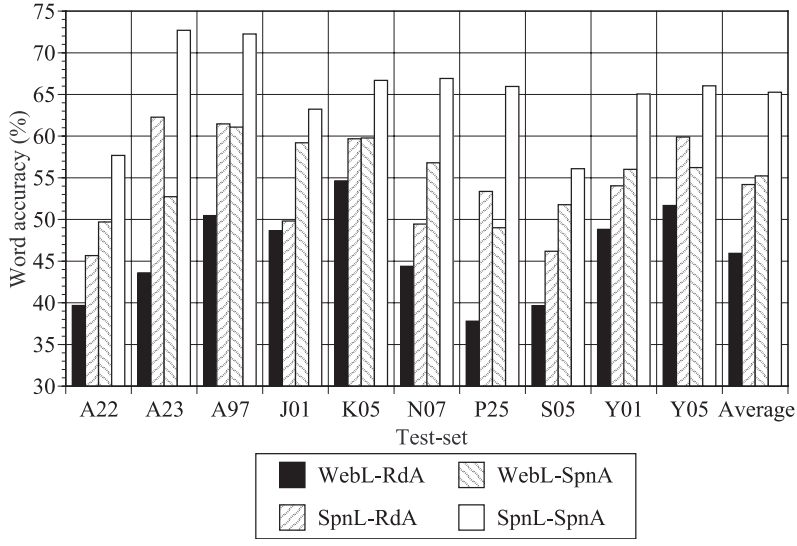

FIGURE 6.7
Word accuracy for each combination of models.

1. Making a regression class tree having 64 leaf nodes for the **SpnA** phone model.
2. Recognizing the test-set utterances using the **SpnA** as a speaker independent model.
3. Applying the MLLR adaptation based on the recognition result for each utterance to make a speaker adaptive model.
4. Re-recognizing the test-set utterances using the speaker adaptive model.
5. Iterating the adaptation process using the resulting transcription.

Figure 6.8 presents the effect of the adaptation when **SpnL** is used as the language model. "SI" indicates the baseline condition using the speaker independent phone model **SpnA**. "SAn" indicates the result after n iterations of the MLLR adaptation. The single step of MLLR improves word accuracy by an absolute 2 to 6%, and the second adaptation step further improves accuracy by 0.7% on average. The improvement almost saturates at the third iteration, and the mean word accuracy after the third iteration is 70.5%. By applying two or three steps of MLLR adaptation, the error rate is reduced by 15% relative to the speaker independent case.

6.5.4 Analysis on Individual Differences

Individual differences in spontaneous presentation speech recognition performances have been analyzed using 10 minutes from each presentation given by 51 male speak-

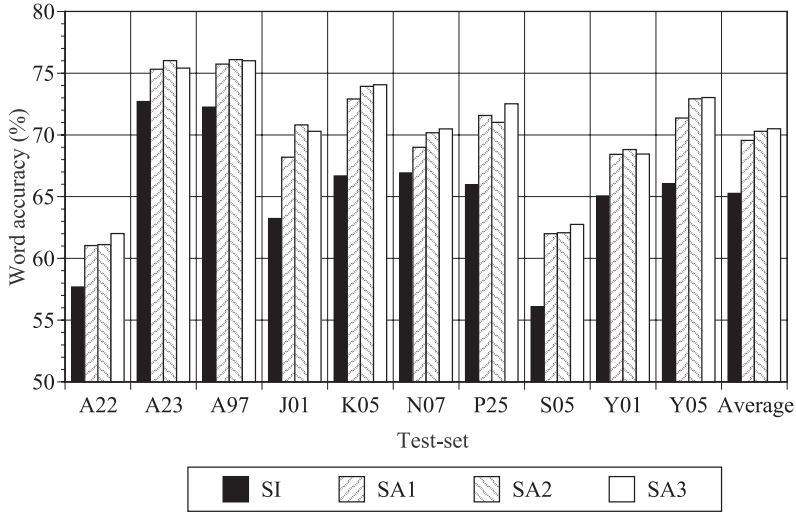

FIGURE 6.8
Results of unsupervised adaptation.

ers, for a total of 510 minutes. The speakers have no overlap with those in the training set. The mean word accuracy for the 51 speakers is 64.1% and 68.6% for the SI (speaker-independent) and SA (speaker-adaptive) conditions respectively. The standard deviation is 7.4% for the SI and 7.5% for the SA condition. As shown by the standard deviation, recognition accuracy largely varies from speaker to speaker. Correlation and regression analyses have been applied to the word recognition accuracy and various speaker attributes.

6.5.4.1 Speaker Attributes

Seven kinds of speaker attributes have been considered in the analysis. They are word accuracy (Acc), averaged acoustic frame likelihood (AL), speaking rate (SR), word perplexity (PP), out of vocabulary rate (OR), filled pause rate (FR), and repair rate (RR). The speaking rate defined as the number of phonemes per second and the averaged acoustic frame likelihood are calculated using the result of forced alignment of the reference tri-phone labels after removing pause periods. The word perplexity is calculated using trigrams, in which prediction of out of vocabulary words is not included. The filled pause rate and the repair rate are the number of filled pauses, and repairs divided by the number of words, respectively. Tag information included in the CSJ transcription is used to determine whether a word is a filled pause/repair or not. In the CSJ, repairs are defined only for word fragments, and a rephrased whole word is not marked as a repair. The calculations of word accuracy, out of vocabulary rate and word perplexity are based on the reference text after excluding repairs.

TABLE 6.3
Correlation coefficient matrix; the lower triangular matrix shows the correlation coefficients and the upper triangular matrix shows the p-value, that is, the significance level. Bold face indicates a significant value with the significance level of 5%.

	Acc(SI)	Acc(SA)	AL(SI)	AL(SA)	SR	PP	OR	FR	RR
Acc(SI)		–	4.7%	–	0.2%	0.4%	0.0 %	0.6%	3.4%
Acc(SA)	–		–	2.2%	0.1%	1.9%	0.0 %	0.5%	2.5%
AL(SI)	**0.28**	–		–	0.0%	55.8%	12. 0%	6.8%	53.9%
AL(SA)	–	**0.32**	–		0.0%	56.3%	8.3 %	6.7%	33.4%
SR	**-0.42**	**-0.47**	**-0.54**	**-0.62**		92.0%	1.7 %	0.0%	20.2%
PP	**-0.40**	**-0.33**	-0.08	-0.08	-0.01		0.0 %	20.0%	69.4%
OR	**-0.54**	**-0.51**	-0.22	-0.25	**0.33**	**0.52**		0.3%	66.5%
FR	**0.38**	**0.38**	0.26	0.26	**-0.50**	-0.18	**-0.41**		33.8%
RR	**-0.30**	**-0.31**	-0.09	-0.14	0.18	0.06	-0. 06	0.14	

6.5.4.2 Correlation Analysis

Table 6.3 shows the correlation matrix of speaker attributes. In the table, the lower triangular matrix shows the correlation coefficients and the upper triangular matrix shows the observed significance levels (p-values). The correlation coefficients written in bold face indicate significant values at 5% significance level (p-values < 0.05).

1. Correlation between acoustic likelihood and speaking rate

   The correlation coefficients between acoustic likelihood and speaking rate are -0.54 and -0.62 for the SI and SA acoustic model, respectively. There is a tendency that the higher the speaking rate is, the lower the acoustic likelihood becomes. The Akaike Information Criterion (AIC) [23] indicates that the first order regression model is better than the second order model for regressing the acoustic likelihood on the speaking rate. This indicates that there is a linear relationship between the speaking rate and the acoustic likelihood averaged over presentations. A stronger articulation effect in faster speakers is probably a cause of the decrease of likelihood. The unsupervised adaptation increases the acoustic likelihood but preserves the relationship between the speaking rate and the acoustic likelihood with a slight increase in the correlation coefficient.

2. Correlation between word perplexity and several linguistic attributes

   There exists significant correlation between the word perplexity and the out of vocabulary rate with a correlation coefficient of 0.52. There is a tendency that presentations having a higher out of vocabulary rate show a higher perplexity. The correlation coefficient of the filled pause frequency, and the perplexity is -0.18 indicating that they are rather uncorrelated. The repair frequency and the perplexity have a correlation coefficient of 0.06. Since the perplexity was calculated after removing repairs, this result shows that the linguistic difficulty excluding repairs has almost no correlation with the repair rate.

3. Correlation between word accuracy and several attributes

   The correlation coefficient between the word accuracy (SI) and the speaking rate is -0.42. Figure 6.9 shows the relationship between the word accuracy and the speaking rate. The relationship seems monotonic and even very slow speaking rate does not decrease the accuracy, which is similar to the result for the acoustic likelihood. The AIC also indicates that the first order model is superior to the second order model for regressing the word accuracy on the speaking rate.

   The correlation between the word accuracy (SI) and the acoustic likelihood is 0.28. In order to analyze the real correlation, partial correlation is calculated. The resultant correlation coefficient adjusted for the speaking rate is -0.07, which means that the correlation is not statistically significant. In other words, the correlation between the word accuracy and the acoustic likelihood is spurious. On the other hand, partial correlation coefficient between the word accuracy and the speaking rate adjusted for the acoustic likelihood is -0.33, which is significant at a 5% significance level, and partial correlation coefficient between the acoustic likelihood and the speaking rate adjusted for the word accuracy is -0.48, which is significant at a 1% significance level. Similar results are obtained for the SA conditions.

   The correlation coefficient between the word accuracy and the repair frequency is -0.30. There is a weak positive correlation of 0.38 between the word accuracy and the filled pause frequency, but this is also a spurious correlation, since partial correlation coefficient adjusted for the speaking rate is 0.22.

   The correlation coefficient between the word accuracy and the out of vocabulary rate is -0.54. There is a negative correlation of -0.40 between the word accuracy (SI) and the perplexity, but this is also spurious; the partial correlation between the word accuracy and the perplexity adjusted for the out of vocabulary rate is -0.16. Figure 6.10 shows the summary of correlation between all the analyzed attributes.

6.5.4.3 Regression Analysis

The following equations (6.1) and (6.2) show linear regression models of the word accuracy with the six presentation attributes when SI and SA acoustic models are respectively used for speech recognition.

$$\begin{aligned} Acc_{SI} &= 0.12AL_{SI} - 0.88SR_{SI} - 0.020PP \\ &\quad -2.2OR + 0.32FR - 3.0RR + 95 \end{aligned} \tag{6.1}$$

$$\begin{aligned} Acc_{SA} &= 0.024AL_{SA} - 1.3SR_{SI} - 0.014PP \\ &\quad -2.1OR + 0.32FR - 3.2RR + 99 \end{aligned} \tag{6.2}$$

In the equation (6.1), regression coefficient for the repair rate is -3.0, and the coefficient for the out of vocabulary rate is -2.2. This means that 1% increase of the repair rate or the out of vocabulary rate respectively corresponds to 3.0% or 2.2% decrease

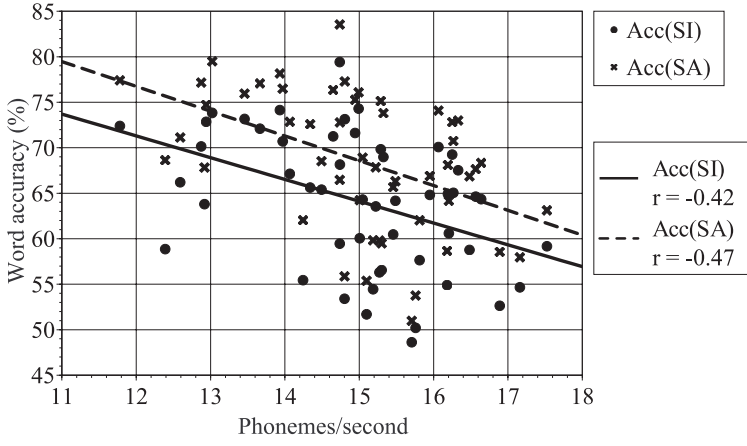

FIGURE 6.9
Speaking rate vs. word accuracy.

of the word accuracy. This is probably because a single recognition error caused by a repair or an out of vocabulary word triggers secondary errors due to the linguistic constraints. The determination coefficients of the multiple linear regressions (6.1) and (6.2) are 0.48 and 0.47, respectively, both of which are significant at 1% level. This means that about half of the variance of the word accuracy can be explained by the model.

Table 6.4 shows normalized representation of the regression analysis with the equations (6.1) and (6.2), in which the variables are normalized in terms of the mean and variance before the analysis in order to show the effects of explaining variables on the word accuracy. The table shows the normalized regression coefficient, the p-value and the 95% confidence interval. The normalized regression coefficients of the speaking rate, the out of vocabulary rate and the repair rate are relatively large especially when SA acoustic model is used. The acoustic likelihood has relatively a small coefficient in both the SI and SA regression models. This means that, although the acoustic likelihood has significant correlation with the word accuracy, it is spurious as indicated in the previous subsection.

6.5.4.4 Selection of Major Attributes

As a supplementary experiment, a backward elimination procedure has been employed to identify relatively important predictors of the word accuracy. A backward elimination process begins with all of the six predictors in the model, and the model is refitted to the data after removing a variable with the largest p-value. The refitting process is iterated removing the least significant variable in the model until all remaining variables have p-values smaller than 0.10. The important predictors identified are the speaking rate, the out of vocabulary rate, and the repair rate, which

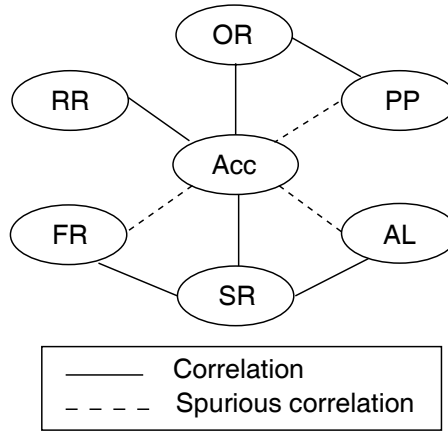

FIGURE 6.10
Summary of correlation between various attributes.

TABLE 6.4
Results of standardized regression analysis for word accuracy, showing standardized regression coefficient (Coeff), p-value and 95% confidence interval (95% CI).

	Coeff(SI)	P	95% CI		Coeff(SA)	P	95% CI
AL(SI)	0.04	76.7%	(-0.22, 0.30)	AL(SA)	0.01	96.1%	(-0.28, 0.29)
SR(SI)	-0.16	32.8%	(-0.47, 0.16)	SR(SI)	-0.23	19.0%	(-0.57, 0.12)
PP	-0.17	20.1%	(-0.44, 0.10)	PP	-0.11	40.4%	(-0.39, 0.16)
OR	-0.34	2.2%	(-0.63,-0.05)	OR	-0.32	3.2%	(-0.62,-0.03)
FR	0.16	24.9%	(-0.11, 0.43)	FR	0.16	26.0%	(-0.12, 0.44)
RR	-0.30	1.4%	(-0.53,-0.06)	RR	-0.31	1.3%	(-0.54,-0.07)

correspond to the attributes showing relatively large coefficients in Table 6.4. The determination coefficients of the regression models on these three attributes are 0.44 for both speaker independent and adaptive cases. This value is almost the same as that of the models on all attributes. It can be concluded that the main factors of individual differences of the word accuracy are the speaking rate, the out of vocabulary rate, and the repair rate.

6.5.5 Discussion

Preliminary recognition experiments have been performed using ten speakers' presentation utterances of approximately 4.5 hours. Language models based on a spontaneous speech corpus and Web corpus were compared in terms of test-set perplexity, OOV rate, and word (morpheme) accuracy. Two acoustic models made by using spontaneous speech and read speech were also compared. Both comparisons showed that acoustic and language modeling based on an actual spontaneous speech corpus is far more effective than conventional modeling based on read speech. It was con-

firmed that the recognition accuracy had a wide speaker-to-speaker variability. When linguistic and acoustic models made form spontaneous speech were used, an average word recognition accuracy of 65.3% was achieved. This performance improved to 70.5% with the help of unsupervised MLLR adaptation for the acoustic model.

Individual differences in the spontaneous presentation speech recognition performances have been investigated using presentations by 51 speakers. A restricted set of the speaker attributes comprising the speaking rate, the out of vocabulary rate, and the repair rate was found to be the most significant to yield individual differences in the word accuracy. The averaged acoustic likelihood of reference phoneme sequences and the test set perplexity were found to be relatively minor factors of individual differences in the word accuracy. Unsupervised MLLR speaker adaptation does not change the structure of the individual differences. Approximately half of the variance in the word accuracy was explained by a regression model using those three major attributes. Future research includes the investigation of efficient methods for reducing the effects of the major attributes on the recognition accuracy. To cope with the speaking rate problem, a method using separate acoustic models for each speaking rate [24] and another method which takes into account the speaking rate in the tree-based HMM state clustering have been proposed [25].

Since the recognition accuracy for spontaneous speech is still rather low, it is imperative to continue the collection of a large corpus of spontaneous speech and use it for building language and acoustic models. Future research issues include: a) how to transcribe and annotate spontaneous speech; b) how to apply morphological analysis to the transcribed spontaneous speech; c) how to build precise and yet general filled pause models; d) how to incorporate repairs, hesitations, repetitions, partial words, and disfluencies; e) how to adapt the language models to each task; f) how to adapt to speaking styles and topics of presentations; and g) how to build acoustic models that fit spontaneous speech.

Segmentation of spontaneous utterances into sentences is one of the important issues. The Viterbi decoding algorithm usually used in speech recognition determines a recognition hypothesis only after detecting the end of the input utterance. In addition, the multiple-pass search algorithm widely used in LVCSR always needs to interrupt the input at some reasonable positions. However, in spontaneous speech, utterances are not separated sentence by sentence. Instead, long pauses are sometimes inserted in a sentence. On the other hand, multiple sentences are sometimes uttered continuously without inserting clear pauses. Therefore, it is necessary to successively determine recognition results before detecting sentence boundaries [10] or inherit a word history for linguistic likelihood calculation to the next sentence hypothesis [26]. In the supporting systems for making presentation records, it is crucial to obtain the N-best hypotheses efficiently, since multiple hypotheses are necessary for error correction in the post processing. For this reason, a new decoder which can process speech continuously without relying on sentence boundary information has been proposed [27].

6.6 Automatic Speech Summarization and Evaluation

6.6.1 Summarization of Each Sentence Utterance

Currently various new applications of LVCSR systems, such as automatic closed captioning [9, 10], making minutes of meetings and conferences [18, 28], and summarizing and indexing of speech documents for information retrieval [29, 30], are actively being investigated. Transcribed speech usually includes not only redundant information such as disfluencies, filled pauses, repetitions, repairs and word fragments, but also irrelevant information caused by recognition errors. Therefore, especially for spontaneous speech, practical applications using speech recognizer require a process of speech summarization which removes redundant and irrelevant information and extracts relatively important information corresponding to users' requirements. Speech summarization producing understandable and compact sentences from original utterances can be considered as a kind of speech understanding. A method for automatically summarizing speech based on word extraction has been investigated at TIT [30, 31]. The method can be applied to the summarization of each sentence/utterance and also to a set of multiple sentences. This subsection explains the case of sentence-by-sentence summarization and its extension to the multiple utterance case is explained in the next subsection. The basic idea of this method is to extract a set of words maximizing a summarization score from an automatically transcribed sentence according to a target compression ratio and recreate a sentence. This method aims to effectively reduce the number of words by removing redundant and irrelevant information without losing relatively important information. The summarization score indicating the appropriateness of a summarized sentence consists of a word significance score I as well as a confidence score C for each word of the original sentence, a linguistic score L for the word string in the summarized sentence, and a word concatenation score T_r. The word concatenation score indicates a word concatenation probability determined by a dependency structure in the original sentence given by a stochastic dependency context free grammar (SDCFG). The total score is maximized using a dynamic programming (DP) technique.
Given a transcription result consisting of N words, $W = w_1, w_2, \cdots, w_N$, the summarization is performed by extracting a set of $M (M < N)$ words, $V = v_1, v_2, \cdots, v_M$, which maximizes the summarization score given by (6.3).

$$S(V) = \sum_{m=1}^{M} \{ L(v_m | \cdots v_{m-1}) + \lambda_I I(v_m) + \lambda_C C(v_m) + \lambda_T T_r(v_{m-1}, v_m) \} \quad (6.3)$$

where λ_I, λ_C, and λ_T are weighting factors for balancing among L, I, C, and T_r.

6.6.1.1 Word Significance Score

The word significance score $I(v_m)$ indicates the relative significance of each word in the original sentence. The amount of information based on the frequency of each

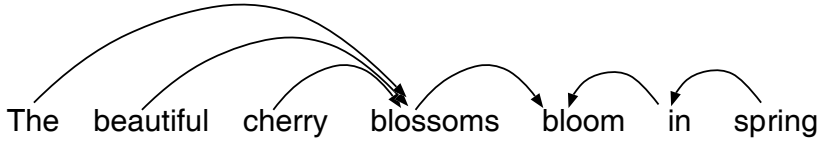

FIGURE 6.11
An example of dependency structure.

word is used as the word significance score for each topic word. We choose nouns and verbs as topic words. A flat score is given to words other than topic words. To reduce the repetition of words in the summarized sentence, a flat score is also given to each reappearing noun and verb.

6.6.1.2 Linguistic Score

The linguistic score $L(v_m|\cdots v_{m-1})$ measured by a bigram probability $P(v_m|v_{m-1})$ indicates the appropriateness of word strings in a summarized sentence.

6.6.1.3 Word Confidence Score

The confidence score $C(v_m)$ is incorporated to weigh acoustically as well as linguistically reliable recognition results. Specifically, a posterior probability of each transcribed word, that is the ratio of a word hypothesis probability to that of all other hypotheses, is calculated using a word graph obtained by a decoder and used as a confidence measure.

6.6.1.4 Word Concatenation Score

Suppose "the beautiful cherry blossoms bloom in spring" is summarized as "the beautiful spring". The latter phrase is a grammatically correct but semantically incorrect summarization. Since the above linguistic score is not powerful enough to avoid such a problem, the word concatenation score $T_r(v_{m-1}, v_m)$ is incorporated to give a penalty for a concatenation between words with no dependency in the original sentence.

The word concatenation in a summarized sentence is restricted by the dependency structure in the original sentence as exemplified in Figure 6.11. The word at the beginning of an arrow is named "modifier" and the word at the end of the arrow is named "head," respectively. The English dependency grammar consists of both "right-headed" dependency indicated by right arrows and "left-headed" dependency indicated by left arrows as shown in Figure 6.11. The dependencies can be written

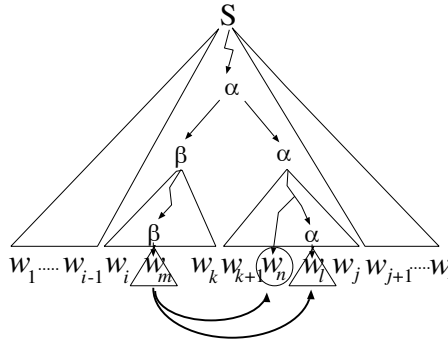

FIGURE 6.12
A phrase structure tree based on a dependency structure.

as phrase structure grammar, DCFG (dependency context free grammar):

$$\alpha \rightarrow \beta\alpha \quad \texttt{(right-headed)}$$
$$\alpha \rightarrow \alpha\beta \quad \texttt{(left-headed)}$$
$$\alpha \rightarrow w$$

where α, β are nonterminal symbols, and w is a terminal symbol (word).

Since the dependencies between words are usually ambiguous, whether dependencies exist or not between words is given by probabilities that one word is modified by others based on the SDCFG. The word dependency probability is a posterior probability estimated by the Inside-Outside probabilities obtained using a manually parsed corpus.

Figure 6.12 illustrates an example of a phrase structure tree based on a dependency structure for a sentence consisting of L words, $w_1, \ldots, w_L$. The probability that w_m and w_l has a dependency structure is calculated as a product of the probabilities of the following sequence when a sentence is derived from the initial symbol S; 1) the rule of $\alpha \rightarrow \beta\alpha$ is applied, 2) $w_i \ldots w_k$ is derived from β, 3) w_m is derived from β, 4) $w_{k+1} \ldots w_j$ is derived from α, and 5) w_l is derived from α. The probability of applying the rule of $\alpha \rightarrow \alpha\beta$ is also added.

In general, as shown in Figure 6.12, a modifier derived from β can be directly connected with a head derived from α in a summarized sentence. In addition the modifier can be also connected with each word which modifies the head. The word concatenation probability between w_m and w_n is defined as a sum of the dependency probabilities between w_m and w_n, and between w_m and each of $w_{n+1} \ldots w_l$. Using the dependency probabilities $d(w_m, w_l, i, k, j)$, the word concatenation score is

calculated by

$$T_r(w_m, w_n) = \log \sum_{i=1}^{m} \sum_{k=m}^{n-1} \sum_{j=n}^{L} \sum_{l=n}^{j} d(w_m, w_l, i, k, j). \tag{6.4}$$

In the SDCFG, only the number of non-terminal symbols is determined and all combinations of rules are applied recursively. The non-terminal symbol has no specific function such as a noun phrase. Even if transcription results by a speech recognizer are ill-formed, the dependency structure can be robustly estimated by the SDCFG.
In the case of Japanese utterance summarization, the word concatenation score is more compact than English, since Japanese sentences have only "right-headed" dependencies. In addition, the word dependency structure in each phrase is deterministic and can be represented by the regular grammar.

6.6.2 Summarization of Multiple Utterances

The automatic speech summarization technique for each sentence has been extended to summarize a set of multiple utterances (sentences) [32]. A set of words maximizing the summarization score is extracted from multiple utterances under some restrictions applied at the sentence boundaries. These restrictions realize the summarization of multiple utterances by handling them as a single long utterance. This results in preserving more words inside information rich utterances and shortening or even completely deleting less informative ones. This summarization technique can be interpreted as a combination of the summarization method extracting important sentences investigated in the field of natural language processing and the sentence-by-sentence summarization method.
Given a transcription result consisting of J utterances, $S_1, \cdots, S_J$ with $(S_j = w_{j1}, w_{j2}, \cdots, w_{jN_j})$ the summarization is performed by extracting a set of $M (M < \sum_j N_j)$ words, $V = v_1, v_2, \cdots, v_M$, which maximizes the summarization score given by equation (6.3).
The amount of calculation for selecting the best combination among all possible combinations of words in the multiple utterances increases as the number of words in the original utterances increases. In order to alleviate this problem, a new method has been proposed, in which each utterance is summarized according to all possible summarization ratios, and then the best combination of summarized sentences for each utterance is determined according to a target compression ratio using a two-level DP technique.

6.6.3 Evaluation

6.6.3.1 Word Network of Manual Summarization Results for Evaluation

To automatically evaluate summarized sentences, correctly transcribed speech is manually summarized by human subjects and used as correct targets. The manual summarization results are merged into a word network which approximately ex-

presses all possible correct summarization including subjective variations. A "summarization accuracy" of automatic summarization is calculated using the word network. A word string that is the most similar to the automatic summarization result extracted from the word network is considered as a correct target for the automatic summarization. The accuracy, comparing the summarized sentence with the target word string, is used as a measure of the linguistic correctness and maintenance of original meanings of the utterance.

6.6.3.2 Evaluation Data

First, Japanese TV broadcast news utterances recorded in 1996 were used to evaluate the proposed method. Fifty utterances with word recognition accuracy above 90%, which was the average rate over the 50 utterances, were selected and used for the evaluation. In addition, five news articles consisting of five sentences each were summarized using the summarization technique for multiple utterances.

Next, English TV broadcast news utterances (CNN news) recorded in 1996 provided by NIST as a test set of topic detection and tracking (TDT) were tagged by the Brill-tagger and used to evaluate the proposed method. Five news articles consisting of 25 utterances in average were transcribed by the JANUS [28] speech recognition system. The multiple utterance summarization was performed for each of the five news articles. Fifty utterances arbitrarily chosen from the five news articles were used for the sentence by sentence summarization. Mean word recognition accuracies of the utterances used for the multiple utterance summarization and those for sentence by sentence summarization were 81% and 80% respectively.

6.6.3.3 Training Data for Summarization Models

Japanese broadcast-news manuscripts recorded from August 1992 to May 1996, comprising of approximately 500k sentences with 22M words, were used both in building a language model for speech recognition and calculating the word significance measure for summarization. A bigram language model for summarization was built using texts from the Mainichi newspaper published from 1996 to 1998, comprising of 5.1M sentences with 87M words. The newspaper text is usually more compact and simpler than broadcast news text and therefore more appropriate for building language models for summarization. Preliminary experiments confirmed that the automatically summarized sentences using word bigram based on newspaper text were much better than those based on broadcast news manuscripts [30]. SDCFG for word concatenation score was built using text from the manually parsed corpus of the Mainichi newspaper published from 1996 to 1998, comprising approximately 4M sentences with 68M words. The number of non-terminal symbols was 100.

In the English speech case, a word significance model, a bigram language model and SDCFG were constructed using roughly 35M words from over 10k sentences of the Wall Street Journal corpus and the Brown corpus in Penn Treebank.

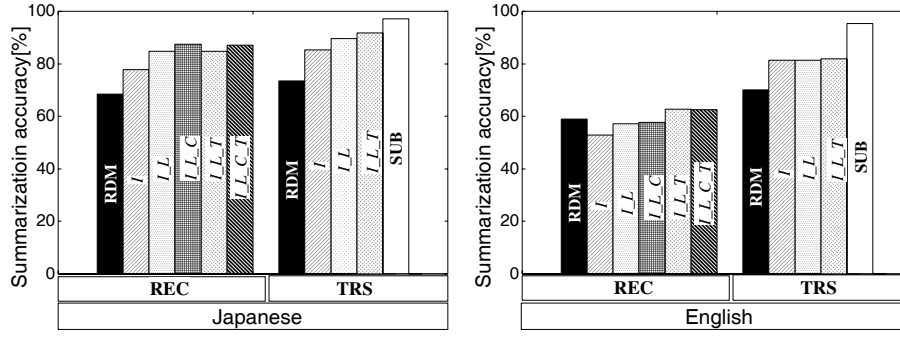

FIGURE 6.13
Each utterance summarizations at 70% summarization ratio.

6.6.3.4 Evaluation Results

Manual transcription (TRS) and automatic transcription (REC) were both summarized. In the summarization of REC, the following score conditions were compared;

- Significance score: I
- Significance and linguistic scores: I_L
- Significance, linguistic, and confidence scores: I_L_C
- Significance, linguistic, and concatenation scores: I_L_T
- All scores: $I\_L\_C\_T$

In the summarization of TRS, since there is no recognition error, the conditions including the confidence score were not tried.
To set the upper limit of the automatic summarization, manual summarization by human subjects for manual transcription (TRS_SUB) was performed. The results were evaluated using all other manual summarization results as correct summarization. In addition, as the upper bound of automatic speech summarization for transcription including speech recognition errors, manual summarization of automatically transcribed utterances was also evaluated (REC_SUB). To ensure that the proposed method is sound, randomly generated summarization sentences were made (RDM) according to the summarization ratio and compared with those obtained by the proposed method.
Figure 6.13 shows results of utterance summarization at 70% summarization ratio for Japanese and English speech, respectively. Figure 6.14 shows those of summarizing articles having multiple sentences at 30% summarization ratio. These results show that the proposed automatic speech summarization technique is significantly more effective than RDM. The better results obtained by incorporating each score indicate that all of the scores are effective to improve the summarization accuracy. Detailed

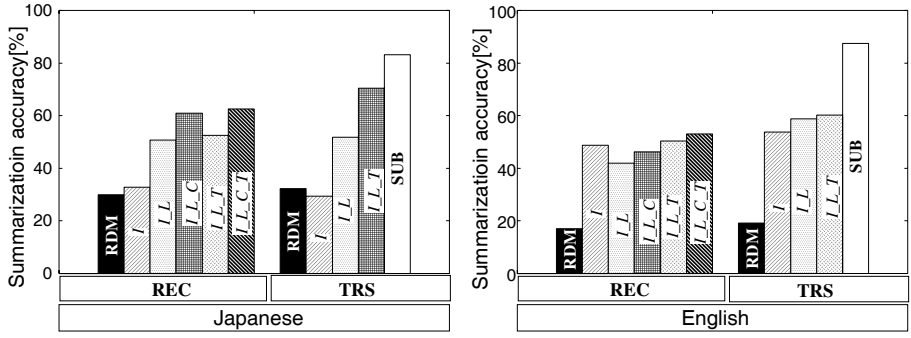

FIGURE 6.14
Article summarizations at 30% summarization ratio.

investigation reveals that the method using the word concatenation score reduces meaning alteration.

6.6.4 Discussion

Each utterance and a whole news article consisting of multiple utterances of Japanese and English broadcast news speech have been summarized by the automatic speech summarization method based on the word significance, linguistic, word confidence, and word concatenation scores. A word set maximizing the total score is extracted by using a dynamic programming technique and connected to build a summarized sentence. A method for measuring the summarization accuracy based on a word network constructed using manual summarization results has also been proposed. Experimental results show that the proposed method can effectively extract relatively important information and remove redundant and irrelevant information from Japanese as well as English news speech. In contrast with the confidence score which has been incorporated into the summarization score to exclude word recognition errors, the linguistic score is effective to reduce out-of-context word extraction both from recognition errors and human disfluencies. In summarizing Japanese news speech, the confidence measure could improve the summarizing performance by excluding in-context word errors. In the English case, the confidence measure can not only exclude word errors but also help extracting clearly pronounced important words. Consequently, the use of the confidence measure yields a larger increase in the summarization accuracy for English than Japanese.

The summarization method is now being applied to the recognition output of presentations recorded in the Japanese national project. Future research includes task-dependent evaluation from the viewpoint of how much the original meaning is maintained in the summarization results based on the performance of IR.

Speech summarization will be applicable to a range of applications, such as making abstracts of presentations, preparing minutes of meetings and voicemails, close cap-

tioning of broadcast news, and presenting information in news-on-demand systems.

6.7 Spontaneous Speech Recognition and Understanding Research Issues

6.7.1 Language Models and Corpora

One of the most important present issues for spontaneous speech recognition is how to create language models (rules). When recognizing spontaneous speech, it is necessary to deal with variations that are not encountered when recognizing speech that is read from texts. These variations include extraneous words, out-of-vocabulary words, ungrammatical sentences, disfluency, partial words, repairs, hesitations, and repetitions. Stochastic language modeling, such as bigrams and trigrams, has been a very powerful tool, so it would be very effective to extend its utility by incorporating semantic knowledge. It would also be useful to integrate unification grammars and context-free grammars for efficient word prediction. It is crucial to develop robust and flexible decoding algorithms that match the characteristics of spontaneous speech.

A paradigm shift from the present transcription-based approach to a detection-based approach will be important to solve the spontaneous-speech specific problems [5]. How to extract contextual information, predict users' responses, and focus on key words are very important issues. Style shifting is also an important problem in spontaneous speech recognition. In typical laboratory experiments, speakers are reading lists of sentences rather than trying to accomplish a real task. Users actually trying to accomplish a task, however, use a different linguistic style. Adaptation of linguistic models, according to tasks and topics, is also a very important issue, since collecting a large linguistic database for every new task is difficult and costly.

The appetites of today's statistical speech processing techniques for training material are well described by the aphorism: "There's no data like more data." Large structured collections of speech and text are essential to progress in speech recognition research. Unlike the traditional approach, in which knowledge of the speech behavior is "discovered" and "documented" by human experts, statistical methods provide an automatic procedure to "learn" the regularities in the speech data directly. The need of a large set of good training data is, thus, more critical than ever. Establishing a good speech database for the machine to uncover the characteristics of the signal is not trivial. There are basically two broad issues to be carefully considered: one being the content and its annotation, and the other the collecting mechanism.

For natural dialog applications such as the ATIS program, a wizard setup is often used to collect the data. A wizard in this case is a human mimicking the machine in interacting with the user. Through the interaction, natural queries in sentential forms are collected. A committee is called upon to resolve cases that may be ambiguous in certain aspects. While a wizard setup can produce a useful set of data, it lacks

diversity, particularly in situations where the real machine may fail. A human wizard cannot intentionally simulate all types of machine error and thus the recorded data may fail to provide complete information of real human-machine interaction.
The recorded data needs to be verified, labeled, and annotated by people whose knowledge will be introduced into the design of the system through its learning process (i.e. via supervised training of the system after the data has been labeled). Labeling and annotation for spontaneous speech can easily become unmanageable. For example, how do we annotate speech repairs and partial words, how do the phonetic transcribers reach a consensus in acoustic-phonetic labels when there is ambiguity, and how do we represent a semantic notion? Errors in labeling and annotation will result in system performance degradation. How to ensure the quality of the annotated results is thus of a major concern. Research in automating or creating tools to assist the verification procedure is by itself an interesting subject.
Another area of research that has gained interest is a modeling methodology and the associated data collection scheme that can reduce the task dependency. To maximize the performance, one should always strive for data that truly reflects the operating condition. It thus calls for a database collection plan that is consistent with the task. This data collection effort would soon become unmanageable if the system designer has to redo data collection for each and every application that is being developed. It is therefore desirable to design a task-independent data set and a modeling method that delivers a reasonable performance upon first use and can quickly allow in-field trials for further revision as soon as task-dependent data become available. Research result in this area can offer the benefit of a reduced application development cost.

6.7.2 Message-driven Speech Recognition and Understanding

State-of-the-art automatic speech recognition systems employ the criterion of maximizing $P(W|X)$, where $W = w_1, \cdots, w_k$ is a word sequence, and $X = x_1, \cdots, x_T$ is an acoustic observation sequence. This criterion is reasonable for dictating read speech. However, the ultimate goal of automatic speech recognition is to extract the underlying messages of the speaker from the speech signals. Hence we need to model the process of speech generation and recognition as shown in Figure 6.15 [33], where M is the message (content) that a speaker intended to convey. The message M is realized as a word sequence W through a linguistic channel, specified by a probability measure $P(W|M)$. The linguistic channel is probabilistic as there are many ways to express the same message, some more likely than others. The word sequence W then gets realized, through the acoustic channel, as a sequence of acoustic signals X. The acoustic channel $P(X|W)$ introduces variability due to various reasons, including speakers and acoustic environments. No one speaker can repeat exactly the same waveform even uttering the same word, and no two speakers are alike in terms of the configuration of their articulatory apparatus. The sequence of sounds radiated from the mouth of the speaker propagates in acoustic waves through the room. The acoustic wave convolved with the room acoustic response and mixed with the acoustic ambient reaches the microphone and is finally converted into an electric signal. The electric signal propagates through a transmission route (cables, wires, or

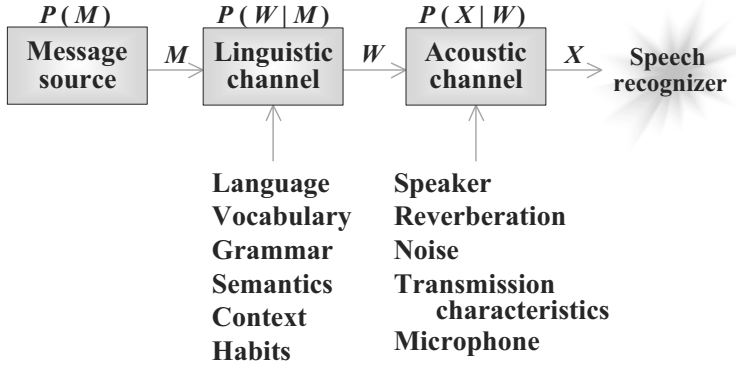

FIGURE 6.15
A communication-theoretic view of speech generation and recognition.

the telephone network) and becomes X when it is received by the recognition and understanding system. Characteristics of all these processes vary substantially.

According to this model, the speech recognition and understanding process is to reverse the generation process to recover M, which can be represented as the maximization of the following a posteriori probability [33]:

$$\max_M P(M|X) = \max_M \sum_W P(M|W)P(W|X). \tag{6.5}$$

Using Bayes' rule, Eq. (1) can be expressed as

$$\max_M P(M|X) = \max_M \sum_W \frac{P(X|W)P(W|M)P(M)}{P(X)}. \tag{6.6}$$

For simplicity, we can approximate the equation as

$$\max_M P(M|X) = \max_{M,W} \frac{P(X|W)P(W|M)P(M)}{P(X)}. \tag{6.7}$$

$P(X|W)$ is calculated using hidden Markov models in the same way as in usual recognition processes. This new formulation of speech recognition was applied to the Japanese broadcast news transcription, and it was found that word error rates were slightly reduced by this method.

There is also a possibility to give feedback from the "understanding module" to the speech recognition module such that decoding hypotheses can be properly adjusted and, hopefully, converge to the most correct word sequence as well as the most correct understanding of the utterance.

6.7.3 Statistical Approaches and Speech Science

There is no doubt that most recent progress in speech recognition comes from statistical approaches, such as HMMs and stochastic language modeling. These approaches were made possible by the recent remarkable progress in computing power. Statistical approaches are usually more reliable and, in many cases, more powerful than knowledge-based approaches, provided that we can obtain a large enough corpus. However, there is always some limit to the size of the corpus and we always encounter some mismatch between the training corpus and the testing data, especially for spontaneous speech. Therefore, even the statistical approaches must be based on reasonable models which can only be created by observing actual phenomena with our knowledge of speech science.

To solve various problems, it is necessary to promote sure and steady research and development by grasping the essence of speech phenomena, instead of developing methods by simply looking at the problems superficially. Speech technology is related to many scientific and engineering fields, including physiology and psychology of speech production and perception, acoustics (physics), signal processing, communication and information theory, computer science, pattern recognition and linguistics; it has an inter-disciplinary nature. It can also be said that speech research exists at the boundary between natural science and engineering.

Knowledge and technology from a wide range of areas, including the use of articulatory and perceptual constraints, will be necessary to develop speech technology. For example, when several phonemes or syllables are continuously spoken, as in the case of usual sentence speech, the tongue, jaw, lips, etc. move asynchronously in parallel, and yet with coupled relationships. Current speech analysis techniques, however, represent speech as a simple time series of spectra. It will become necessary to analyze speech by decomposing it into several hidden factors based on speech production mechanisms. This approach seems to be essential for solving the coarticulation problem, one of the most important problems in speech recognition.

The human hearing system is far more robust than machine systems - more robust not only against the direct influence of additive noise but also against speech variations (that is, the indirect influence of noise), even if the noise is very inconsistent. Speech recognizers are therefore expected to become more robust when the front end utilizes models of human hearing. This can be done by imitating the physiological organs or by reproducing psychoacoustic characteristics. Although it is not always necessary or efficient for speech recognition systems to directly imitate human speech production and perception mechanisms, it will become more important in the near future to build mathematical models based on these mechanisms in order to improve the performance of spontaneous speech recognition [34].

6.7.4 Research on the Human Brain

Up to the present, the fields of speech perception and automatic speech recognition have been widely separate. However, in order to build spontaneous speech understanding systems, it is crucial to analyze the function within the human brain. The

function must then be realized using engineering models. For these purposes, human speech perception research needs to shift from targeting short fragments such as phonemes and syllables to larger units such as words, phrases, sentences, and paragraphs. Research should investigate how meanings conveyed by speech are understood. It is indispensable to build a large corpus of spontaneous speech and conduct corpus-based research on both the mechanisms of human speech perception and the engineering speech understanding systems with close connection and cooperation.
Ultimately, in order to make speech recognition systems really useful and comfortable for users, they should match or exceed human capabilities. That is, they should be faster, more accurate, more intelligent, more knowledgeable, less expensive, and easier to communicate with than human staff. For this purpose, the ultimate systems must be able to handle conceptual information. Figure 6.16 shows a diagram of human speech generation and perception process. Although observation and modeling of the movement of vocal systems along with the physiological modeling of auditory peripheral systems have recently made great progress, the mechanism of speech information processing in our human brain has hardly been investigated. Psychological experiments on human memory clearly show that speech plays a far more important and essential role than vision in the human memory and thinking processes. Whereas models of separating acoustic sources have been researched in "auditory scene analysis," the mechanisms of how meanings of speech are understood and how speech is generated have not yet been made clear.
It will be necessary to clarify the processes by which human beings understand and produce spontaneous speech, in order to obtain hints for constructing language models for spontaneous speech, which is very different from written language. It is necessary to be able to analyze and utilize contextual information to handle anaphora and ellipsis frequently used in human dialogues. It is time to start active research on clarifying the mechanism of speech information processing in the human brain so that epoch-making technological progress can be made based on the human model.

6.7.5 Dynamic Spectral Features

Psychological and physiological research into human speech perception mechanisms shows that the human hearing organs are highly sensitive to changes in sounds, i.e., to transitional (dynamic) sounds, and that the transitional features of the speech spectrum and the speech wave play crucial roles in phoneme perception [35]. The length of the time windows in which sound transitions are perceived have a hierarchical structure and range from the order of several milliseconds to several seconds. The hierarchical layers correspond to various speech features, such as phonemes, syllables and prosodic features. It has also been reported that the human hearing mechanism perceives a target value estimated from the transitional information extracted using dynamic spectral features.
The representation of the dynamic characteristics of speech waves and spectra has been studied, and several useful methods have been proposed [36, 37]. However, the performance of these methods is not yet satisfactory and most of the successful speech analysis methods developed thus far assume a stationary signal at least for

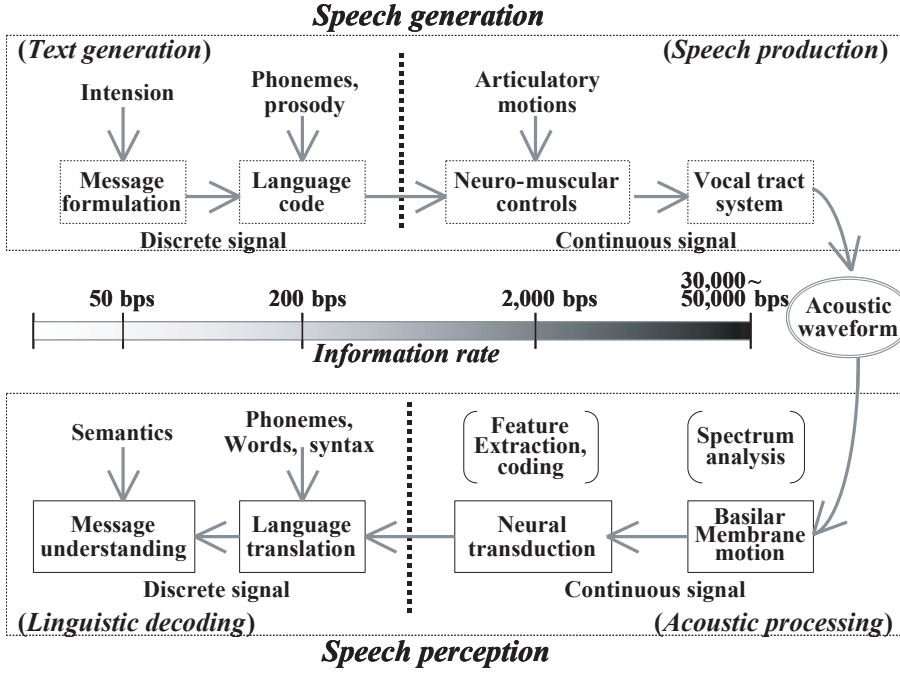

FIGURE 6.16
Speech-generation and speech-perception processes.

each basic short period. It is still very difficult to relate time functions of pitch and energy to perceptual prosodic information. If good methods for representing the dynamics of speech associated with various time lengths are discovered, they should have a substantial impact on the course of spontaneous speech research.

6.8 Conclusion

Although high recognition accuracy can be obtained for speech in the form of reading a written text or similar by using state-of-the art speech recognition technology, the accuracy is quite poor for freely spoken spontaneous speech. This chapter discussed the most important research problems to be solved in order to achieve ultimate spontaneous speech recognition systems, and tried to forecast where progress will be made in the near future. The problems include language and acoustic modeling of spontaneous speech, spontaneous speech corpus building, message-driven speech recognition and understanding, and speech summarization. A paradigm shift from

speech recognition to understanding, where the underlying messages of the speaker, i.e., meaning/content that the speaker intended to convey, are extracted, instead of simply transcribing all the spoken words, will be indispensable.

To meet this need, a five-year national project for raising the technological level of speech recognition and understanding commenced in Japan in 1999. The project focuses on building a large-scale spontaneous speech corpus together with acoustic and linguistic modeling for spontaneous speech recognition and summarization. Experimental results show that acoustic and language modeling based on the actual spontaneous speech corpus is far more effective than modeling based on read speech. It is also shown that the proposed automatic speech summarization method effectively extracts relatively important information and removes redundant and irrelevant information.

It will become important to use articulatory and perceptual constraints to solve various fundamental problems in spontaneous speech modeling. It will also become crucial to analyze the function within the human brain, that is, how human-beings are understanding speech, and the function must then be realized using engineering models. Research should investigate how meanings conveyed by speech are understood.

References

[1] B.-H. Juang and S. Furui: "Automatic recognition and understanding of spoken language - A first step towards natural human-machine communication", Proc. IEEE, 88, 8, pp. 1142-1165 (2000)

[2] L. R. Rabiner and B. H. Juang: Fundamentals of Speech Recognition, New Jersey, Prentice-Hall, Inc. (1993)

[3] S. Furui: Digital Speech Processing, Synthesis, and Recognition, 2nd Edition, New York, Marcel Dekker (2000)

[4] H. Ney: "Corpus-based statistical methods in speech and language processing", in Corpus-based Methods in Language and Speech Processing, Young, S. and Bloothooft, G. Ed., pp. 1-26 (1997)

[5] B. H. Juang: "From speech recognition to understanding: Shifting paradigm to achieve natural human-machine communication", Proc. 16th ICA and 135th Meeting ASA, pp. 617-618 (1998)

[6] J. J. Godfrey, E. C. Holliman and J. McDaniel: "Switchboard: Telephone speech corpus for research and development", Proc. ICASSP, pp. I-517-520 (1992)

[7] A. Waibel, et al.: "Advances in automatic meeting record creation and access", Proc. ICASSP 2001, pp. I-597-600 (2001)

[8] A. I. Rudnicky: “Hub 4: Business broadcast news”, Proc. DARPA Speech Recognition Workshop, pp. 8-11 (1996)

[9] K. Ohtsuki, et al.: “Recent advances in Japanese broadcast news transcription”, Proc. Eurospeech99, pp. 671-674 (1999)

[10] T. Imai, et al.: “Progressive 2-pass decoder for real-time broadcast news captioning”, Proc. ICASSP 2000, pp. III-1559-1562 (2000)

[11] J.-L. Gauvain, L. Lamel, Y. de Karcadio and G. Adda: “Transcription and indexation of broadcast data”, Proc. ICASSP 2000, pp. III-1663-1666 (2000)

[12] J. Hirschberg, M. Bacchiani, D. Hindle, P. Isenhour, A. Rosenberg, L. Stark, L. Stead, S. Whittaker and G. Zamchick: “SCANMail: Browsing and searching speech data by content”, Proc. Eurospeech2001, pp. 1299-1302 (2001)

[13] K. Koumpis, S. Renals and M. Niranjan: “Extractive summarization of voicemail using lexical and prosodic feature”, Proc. Eurospeech2001, pp. 2377-2380 (2001)

[14] A. Aaron, et al.: “Speech recognition for DARPA Communicator”, Proc. ICASSP 2001, pp. I-489-492 (2001)

[15] E. Levin, et al.: “The AT&T-DARPA Communicator mixed-initiative spoken dialog system”, Proc. ICSLP 2000, Beijing, pp. II-122-125 (2000)

[16] L. G. Miller and A. Gorin: “Structured networks for adaptive language acquisition”, Int. J. Pattern Recognit. Artif. Intell. (Special Issue on Neural Networks), 7, 4, pp. 873-898 (1993)

[17] S. Deerwester, et al.: “Indexing by latent semantic analysis”, J. Amer. Soc. Inform. Sci., 41, pp. 391-407 (1990)

[18] S. Furui, K. Maekawa, H. Isahara, T. Shinozaki and T. Ohdaira: “Toward the realization of spontaneous speech recognition - Introduction of a Japanese priority program and preliminary results”, Proc. Int. Conf. Spoken Language Processing, Beijing, pp. III-518-521 (2000)

[19] K. Maekawa, et al.: “Spontaneous speech corpus of Japanese”, Proc. 2nd International Conference on Language Resources and Evaluation, Athens, Greece, pp. 947-952 (2000)

[20] C. J. Legetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models”, Computer Speech and Language, 9, pp. 171-185 (1995)

[21] S. Young, et al.: “The HTK Book, Version 2.2”, Entropic Ltd (1999)

[22] T. Shinozaki, C. Hori and S. Furui: “Towards automatic transcription of spontaneous presentations”, Proc. Eurospeech2001, Aalborg, Vol. 1, pp. 491-494 (2001)

[23] H. Akaike: “Information theory and an extension of the maximum likelihood principle”, Proc. ISIT, (B. N. Petrov and F. Csaki eds.) Akademiai Kiado, Budapest, pp.267-281 (1973)

[24] J. Zheng, H. Franco and F. Weng: “Word-level rate of speech modeling using rate-specific phones and pronunciations”, Proc. ICASSP 2000, pp. 1775-1778 (2000)

[25] C. Fugen and I. Rogina: “Integrating dynamic speech modalities into context decision trees”, Proc. ICASSP 2000, pp. 353-356 (2000)

[26] O. Segawa, T. Takeda and F. Itakura: “Continuous speech recognition without end-point detection”, Proc. ICASSP 2001, pp. I-245-248 (2001)

[27] T. Kawahara, H. Nanjo and S. Furui: “Automatic transcription of spontaneous lecture speech”, Proc. ASRU2001 (2001)

[28] Z. Klaus: “Automatic generation of concise summaries of spoken dialogues in unrestricted domains”, Proc. SIGIR2001, New Orleans (2001)

[29] R. Valenza et al.: “Summarization of spoken audio through information extraction”, Proc. ESCA Workshop on Accessing Information in Spoken Audio, Cambridge, pp. 111-116 (1999)

[30] C. Hori and S. Furui: “Automatic speech summarization based on word significance and linguistic likelihood”, Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Istanbul, pp. 1579-1582 (2000)

[31] C. Hori and S. Furui, “Improvements in automatic speech summarization and evaluation methods”, Proc. Int. Conf. Spoken Language Processing, Beijing, pp. IV-326-329 (2000)

[32] C. Hori and S. Furui, “Advances in automatic speech summarization”, Proc. Eurospeech2001, Aalborg, pp. III-1771-1774 (2001)

[33] K. Ohtsuki, S. Furui, A. Iwasaki and N. Sakurai: “Message-driven speech recognition and topic-word extraction”, Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Phoenix, pp. 625-628 (1999)

[34] S. Furui: “Towards the Ultimate Synthesis/Recognition System”, in Voice Communication between Humans and Machines, eds. by Roe, D. B. and Wilpon, J. G., Washington D. C., National Academy Press, pp. 450-466 (1994)

[35] S. Furui: “On the role of spectral transition for speech perception”, J. Acoust. Soc. Am., 80, 4, pp. 1016-1025 (1986)

[36] S. Furui and M. Akagi: “On the role of spectral transition in phoneme perception and its modeling”, Proc. 12th ICA, Toronto, Canada, A2-6 (1986)

[37] S. Furui: “Speaker-independent isolated word recognition using dynamic features of speech spectrum”, IEEE Trans. Acoust., Speech, Signal Process., ASSP-34, 1, pp. 52-59 (1986)

7

Speaker Authentication

Qi Li* **and Biing-Hwang Juang†**

**Bell Labs; †Avaya Labs Research*

CONTENTS

Among various user authentication techniques, speaker authentication concerns with authenticating a person's identity via voice. There are two approaches to speaker authentication: speaker verification (SV) and verbal information verification (VIV). The SV approach attempts to verify a speaker's identity based on his/her voice characteristics while the VIV approach verifies a speaker's identity through verification of the content of his/her utterance(s). In this chapter, we first introduce the related pattern recognition and verification techniques, and then present an SV system, a VIV system, and a combined system with both SV and VIV for convenience and performance improvement. These systems are ready for real-world applications.

7.1 Introduction

To ensure the security of and proper access to private information, important transactions, and the computer and communication networks, passwords or personal identification numbers (PIN) have been used extensively in our daily life. To further enhance the level of security as well as convenience, biometric features such as signature, fingerprint, hand shape, eye iris, and voice have also been considered. Among all biometric features, a person's voice is the most convenient one for per-

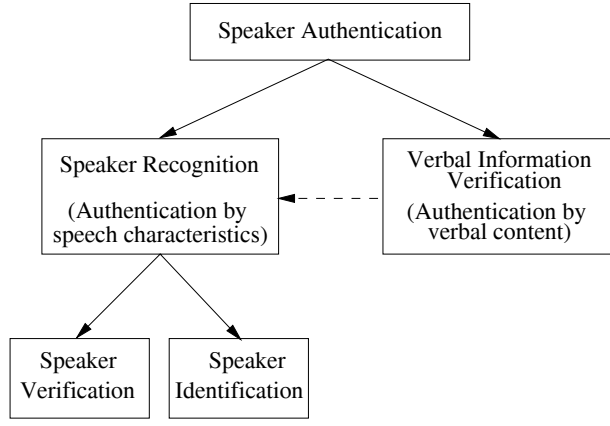

FIGURE 7.1
Speaker authentication approaches.

sonal identification purposes because it is easy to produce, capture, or transmit over the ubiquitous telephone network. It also can be supported with existing services without requiring special devices. *Speaker authentication* as an application of pattern recognition is the process of authenticating a user via his/her spoken input. How to automate speedily the authentication procedure and achieve a high accuracy poses a serious technical challenge to speech researchers.

As shown in Fig. 7.1, the approach to speaker authentication can be categorized into two groups: one uses a speaker's voice characteristics, which leads to speaker recognition and the other focuses on the verbal content of the spoken utterance, which leads to verbal information verification. These two techniques can also be combined to provide an enhanced system as indicated by the dashed line.

7.1.1 Speaker Recognition and Verification

Speaker recognition can be formulated in two operating modes, speaker verification and speaker identification. *Speaker verification* (SV) is the process of verifying an unknown speaker whether s/he is the person as claimed, i.e. a yes-no *hypothesis testing* problem. *Speaker identification* (SID) is the process of associating an unknown speaker with a member in a pre-registered, known population, i.e. a multiple-choice *classification* problem. In this chapter, we will focus on the task of speaker verification.

Speaker recognition as one of the voice authentication techniques has been studied for several decades [2, 3, 40, 6, 22]. A typical SV system is shown in Fig. 7.2, which has two operating scenarios: enrollment and test sessions. A speaker needs to enroll first before s/he can use the system. In the enrollment session, the user's identity, such as an account number, together with a pass-phrase, such as a digit string or a key phrase like "open sesame" shown in the figure, is assigned to the

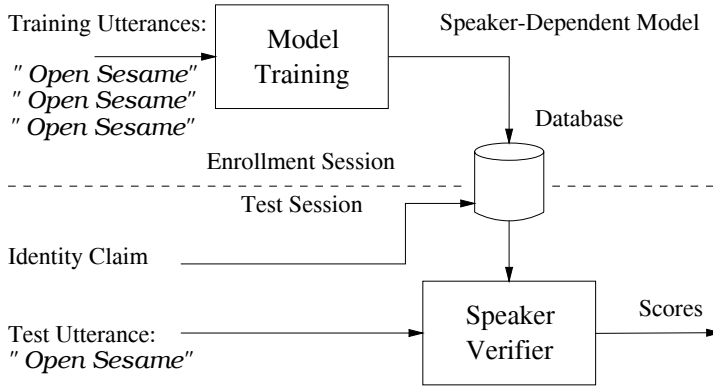

FIGURE 7.2
A speaker verification system.

speaker. The system then prompts the speaker to say the pass-phrase several times to allow training or constructing of a speaker-dependent (SD) model that registers the speaker's speech characteristics. The digit string can be the same as the account number and the key phrase can be selected by the user so it is easy to remember. An enrolled speaker can use the verification system in a future test. (Similar procedure applies in the case of speaker identification. These schemes are sometimes referred to as *direct method* as they use the talker's speech characteristics to infer or verify the talker's identity *directly*.) In a test session, the user first claims his/her identity by entering or speaking the identity information. The system then prompts the speaker to say the pass-phrase. The pass-phrase utterance is compared against the stored SD model. The speaker is accepted if the verification score exceeds a preset threshold; otherwise, the speaker is rejected. Note that the pass-phrase may or may not be kept in secret.

When the pass-phrases are the same in training and testing, the system is called a *fixed pass-phrase system*. Frequently, a short phrase or a connected-digit sequence, such as a telephone or account number, is chosen as the fixed pass-phrase. Using a digit string for a pass-phrase has a distinctive difference from other non-digit choices. The high performance of current connected digit speech recognition systems and embedded error correcting possibilities of digit strings make it feasible that the identity claim can be made via spoken, rather than key-in input [41, 42]. If such an option is installed, the spoken digit string is first recognized by an automatic speech recognizer (ASR) and the standard verification procedure then follows using the same digit string. Obviously, successful verification of a speaker relies upon a correct recognition of the input digit string.

A security concern may be raised about using fixed pass-phrases since a spoken pass-phrase can be tape-recorded by impostors and used in later trials to get access to the system. A text-prompted SV system has been proposed to circumvent such a problem. A *text-prompted system* uses a set of speaker-dependent word or subword mod-

els, possibly for a small vocabulary such as the digits. These models are employed as the building blocks for constructing the models for the prompted utterance, which may or may not be part of the training material. When the user tries to access the system, the system prompts the user to utter a randomly picked sequence of words in the vocabulary. The word sequence is aligned with the pre-trained word models and a verification decision is made based upon the evaluated likelihood score. Compared to a fixed-phrase system, such a text-prompted system normally needs longer enrollment time in order to collect enough data to train the SD word or subword models. The performance of a text-prompted system is, in general, not as high as that of a fixed-phrase system. This is due to the fact that the phrase model constructed from concatenating elementary word or subword models is usually not as accurate as that directly trained from the phrase utterance in a fixed-phrase system. Details on a text-prompted system and its performance can be found (e.g. [26]).

The above systems are called text-dependent, or text-constrained SV systems because the input utterance is constrained, either by a fixed phrase or by a fixed vocabulary. A verification system can also be text-independent. In a *text-independent SV system*, a speaker's model is trained on the general speech characteristics of the person's voice [39, 13]. Once such a model is trained, the speaker can be verified regardless of the underlying text of the spoken input. Such a system has wide applications in monitoring applications for verifying a speaker on a continuous basis. In order to characterize a speaker's general voice pattern without a text constraint, we normally need a large amount of phonetically or acoustically rich training data in the enrollment procedure. Also, without the text or lexical constraint, longer test utterances are usually needed to maintain a satisfactory SV performance. Without a large training set and long test utterances, the performance of a text-independent system is usually inferior to that of a text-dependent system.

In evaluating an SV system, if it is both trained and tested by the same set of speakers, it is called a *closed test*; otherwise, an *open test*. In a closed test, data from all the potential impostors (i.e., all except the true speaker) in the population can be used to train a set of high performance, discriminative speaker models. However, as most SV applications are of an open-test nature, to train the discriminative model against all possible impostors is not possible. As an alternative, a set of speakers whose speech characteristics are close to the speaker can be used to train the SD discriminative model, or speaker independent models can be used to model impostors.

7.1.2 Verbal Information Verification

When applying the current speaker recognition technology to real-world applications, several problems are encountered. One of such problems is the need of an enrollment session to collect data for training the speaker-dependent (SD) model. Enrollment is an inconvenience to the user as well as the system developer who often has to supervise and ensure the quality of the collected data. The quality of the collected training data has a critical effect on the performance of an SV system. A speaker may make a mistake when repeating the training utterances/pass-phrases for several times. Furthermore, as we have discussed in [25], since the enrollment and

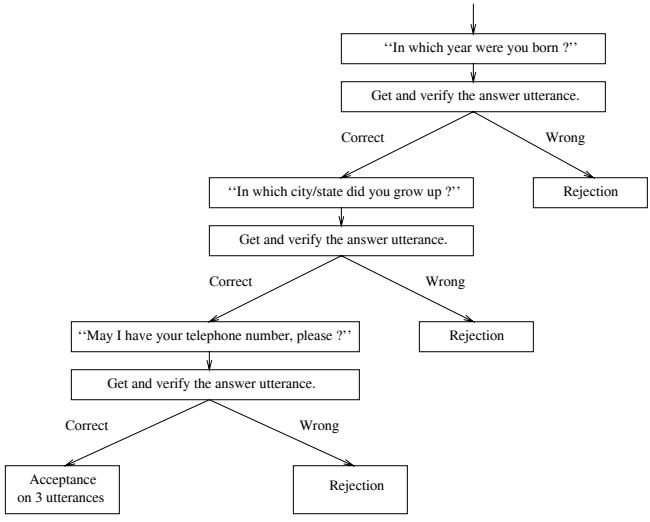

FIGURE 7.3
An example of verbal information verification by asking sequential questions. (Similar sequential tests can also be applied in speaker verification and other biometric or multi-modality verification.)

testing voice may come from different telephone handsets and networks, acoustic mismatch between the training and testing environments may occur. The SD models trained on the data collected in an enrollment session may not perform well when the test session is in a different environment or via a different transmission channel. The mismatch significantly affects the SV performance. This is a significant drawback of the direct method, in which the robustness in comparative evaluation is difficult to ensure. Alternatively, in light of the progress in modeling for speech recognition, the concept and algorithm of VIV was proposed [24] to take advantage of the different characteristic focus on the speech signal, namely, that of the speaker vs. that of the speech.

The VIV method is the process of verifying spoken utterances against the information stored in a given personal data profile. A VIV system may use a dialogue procedure to verify a user by asking questions. An example of a VIV system is shown in Fig. 7.3. It is similar to a typical tele-banking procedure: after an account number is provided, the operator verifies the user by asking some personal information, such as mother's maiden name, birth date, address, home telephone number, etc. The user must provide answers to the questions correctly in order to gain access to his/her account and services. In this manner, a talker's identity is embedded in the knowledge s/he has towards some particular questions and thus one often considers VIV an *indirect method*. To automate the whole procedure, the questions can be prompted by a text-to-speech system (TTS) or by pre-recorded messages.

The difference between speaker recognition (the direct method) and verbal informa-

tion verification (the indirect method) can be further addressed in the following three aspects. First, in a speaker recognition system, either for speaker identification or for speaker verification, we need to train speaker-dependent (SD) models, while in VIV we usually use statistical models with associated acoustic-phonetic identities. Second, a speaker recognition system needs to enroll a new user and to train the SD model while a VIV system does not require voice enrollment. Instead, a user's personal data profile is created when the user's account is set up. Finally, in speaker recognition, the system has the ability to reject an imposter even when the input utterance contains a legitimate pass-phrase, if the utterance indeed fails to match the pre-trained SD model. In VIV, it is solely the user's responsibility to protect his or her personal information because no speaker-specific voice characteristics are used in the verification process. In real applications, there are several ways to circumvent the situation in which an impostor uses a speaker's personal information obtained from eavesdropping a particular session. A VIV system can ask for information that may not be a constant from one session to another, e.g. the amount or date of the last deposit, or a subset of the registered personal information, i.e. a number of randomly selected information fields in the personal data profile. Furthermore, as we are going to present in Section 7.5, a VIV system can be migrated to an SV system as indicated by the dash line in Fig. 7.1. In particular, VIV can be used to facilitate automatic enrollment for SV.

7.2 Pattern Recognition in Speaker Authentication

In this section, we review pattern recognition techniques in speaker authentication. Starting with the Bayesian decision theory, we introduce the statistical modeling approach for stationary and non-stationary processes, algorithms for speech segmentation, and hypothesis testing.

7.2.1 Bayesian Decision Theory

In an M-class recognition problem, we are 1) given an observation (or a feature vector) $\mathbf{o}$ in a d-dimensional Euclidean space $\mathbf{R}^d$, and a set of classes designated as $\{C_1, C_2, ..., C_M\}$, and 2) asked to make a decision, to classify $\mathbf{o}$ into, say, class C_i, where one class can be one speaker or one acoustic unit. We denote this as an action α_i. By Bayes formula, the probability of being class C_i given $\mathbf{o}$ is the posterior (or *a posteriori*) probability:

$$P(C_i|\mathbf{o}) = \frac{p(\mathbf{o}|C_i)P(C_i)}{p(\mathbf{o})} \tag{7.1}$$

where $p(\mathbf{o}|C_i)$ is the conditional probability, $P(C_i)$ is prior probability, and

$$p(\mathbf{o}) = \sum_{j=1}^{M} p(\mathbf{o}|C_j)P(C_j) \tag{7.2}$$

can be viewed as a scale factor that guarantees that the posterior probabilities sum to one.

Let $\mathcal{L}(\alpha_i|C_j)$ be the loss function describing the loss incurred for taking action α_i when the true class is C_j. The expected loss (or risk) associated with taking action α_i is

$$R(\alpha_i|\mathbf{o}) = \sum_{j=1}^{M} \mathcal{L}(\alpha_i|C_j)P(C_j|\mathbf{o}). \tag{7.3}$$

This leads to the *Bayes decision rule*: To minimize the overall risk, compute the above risk for $j = 1, ..., M$ and then select the action α_i such that $R(\alpha_i|\mathbf{o})$ is minimum.

For speaker authentication, we are interested in the zero-one loss function:

$$\mathcal{L}(\alpha_i|C_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j. \end{cases} \qquad i, j = 1, ..., M \tag{7.4}$$

It assigns no loss to a correct decision and a unit loss to an error, equivalent to counting the errors. The risk to this specific loss function is

$$R(\alpha_i|\mathbf{o}) = \sum_{j=1}^{M} \mathcal{L}(\alpha_i|C_j)P(C_j|\mathbf{o}) \tag{7.5}$$

$$= \sum_{j \neq i} P(C_j|o) = 1 - P(C_i|\mathbf{o}). \tag{7.6}$$

Thus, to minimize the risk or error rate, we take action α_k that maximizes the posterior probability $P(C_i|\mathbf{o})$:

$$\text{Take action } \alpha_k, \text{ where } k = \arg \max_{1 \leq i \leq M} P(C_i|\mathbf{o}). \tag{7.7}$$

Since the expected value of this loss function is equivalent to error rate, this is also called minimum-error-rate classification [9]. Recalling the Bayes formula in Eq. (7.1), when the density $p(\mathbf{o}|C_i)$ has been estimated for all classes and the prior probabilities are known, we can rewrite the above decision rule as:

$$\text{Take action } \alpha_k, \text{ where } k = \arg \max_{1 \leq i \leq M} p(\mathbf{o}|C_i)P(C_i). \tag{7.8}$$

So far, we only consider the case of a single observation (or feature vector) $\mathbf{o}$. In speaker authentication, we always encounter or employ a sequence of observations $\mathbf{O} = \{\mathbf{o}_i\}_{i=1}^{T}$, where T is the total number of observations. After speech segmentation (which will be discussed later), we assume that during a short time period

these sequential observations are produced by the same speaker and they belong to the same acoustic class or unit, say C_i. Furthermore, if we assume that the observations are independent and identically distributed (i.i.d.), the joint posterior probability, $P(C_i|\mathbf{O})$ is merely the product of the component probabilities:

$$P(C_i|\mathbf{O}) = \prod_{t=1}^{T} P(C_i|\mathbf{o}_t). \tag{7.9}$$

From Eq. (7.8), the decision rule for the compound decision problem is

$$\alpha_k = \arg \max_{1 \le i \le M} \prod_{t=1}^{T} p(\mathbf{o}_t|C_i)P(C_i). \tag{7.10}$$

In practice, the decision is usually based on the log likelihood score:

$$\alpha_k = \arg \max_{1 \le i \le M} \sum_{t=1}^{T} \log p(\mathbf{o}_t|C_i)P(C_i). \tag{7.11}$$

7.2.2 Stochastic Models for Stationary Process

As discussed above, the decision on authentication is made by computing the likelihood based on the probability density functions (*pdf*s) of the feature vector. Parameters that define these *pdf*s have to be estimated a priori.

There are many model structures general enough to characterize a speech *pdf* . Here we focus on the Gaussian mixture model (GMM), which is defined as:

$$p(\mathbf{o}_t|C_j) = p(\mathbf{o}_t|\lambda_j) = \sum_{i=1}^{I} c_i \mathcal{N}(\mathbf{o}_t; \mu_i, \Sigma_i), \tag{7.12}$$

where λ_j is the GMM for class C_j, c_i is a mixture weight which must satisfy the constraint $\sum_{i=1}^{I} c_i = 1$, I is the total number of mixture components, and $\mathcal{N}(\cdot)$ is a Gaussian density function:

$$\mathcal{N}(\mathbf{o}_t; \mu_i, R_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{o}_t - \mu_i)^{\mathrm{T}} \Sigma_i^{-1} (\mathbf{o}_t - \mu_i)\right\}, \tag{7.13}$$

where μ_i and Σ_i are the d-dimensional mean vector and covariance matrix of the i'th component.

Given a sequence of feature vectors, the GMM parameters can be estimated iteratively using a hill-climbing algorithm such as the Baum-Welch [5] or the expectation-maximization (EM) algorithm [8]. As has been proved, the algorithm ensures monotonic increase in the log-likelihood during the iterative procedure until a fixed-point solution is reached [58, 14]. In most applications, model parameter estimation can

be accomplished in a few iterations. At each step of the iteration, the parameter estimation formulas for mixture i are:

$$\hat{c}_i = \frac{1}{T}\sum_{t=1}^{T} p(i|\mathbf{o}_t, \lambda) \tag{7.14}$$

$$\hat{\mu}_i = \frac{\sum_{t=1}^{T} p(i|\mathbf{o}_t, \lambda)\mathbf{o}_t}{\sum_{t=1}^{T} p(i|\mathbf{o}_t, \lambda)} \tag{7.15}$$

$$\hat{\Sigma}_i = \frac{\sum_{t=1}^{T} p(i|\mathbf{o}_t, \lambda)(\mathbf{o}_t - \hat{\mu}_i)(\mathbf{o}_t - \hat{\mu}_i)^{\mathrm{T}}}{\sum_{t=1}^{T} p(i|\mathbf{o}_t, \lambda)} \tag{7.16}$$

where

$$p(i|\mathbf{o}_t, \lambda) = \frac{p(\mathbf{o}_i|\lambda)c_i}{\sum_{j=1}^{I} p(\mathbf{o}_i|\lambda)c_j}. \tag{7.17}$$

One application of the above model is context-independent speaker identification, where we assume that each speaker's speech characteristics manifest only acoustically and is represented by one (model) class. When a spoken utterance is long enough, it is reasonable to assume that the acoustic characteristic is independent of its content. For a group of M speakers, in the enrollment phase, we train M GMM's, $\lambda_1, \lambda_2, ..., \lambda_M$, using the reestimation algorithm respectively. In the test phase, given an observation sequence $\mathbf{O}$, the objective is to find in the prescribed speaker population the speaker model that achieves the maximum posterior probability. From Eq. (7.11), assume the prior is the same for all speakers, the decision rule is

$$\text{Take action } \alpha_k, \text{ where } k = \arg\max_{1\le i\le M} \sum_{t=1}^{T} \log p(\mathbf{o}_t|\lambda_i). \tag{7.18}$$

where α_k is the action of deciding that the observation $\mathbf{O}$ is from speaker k.
The vector quantization (VQ) method [46] is another approach to speaker identification. It uses a *speaker-dependent* codebook to characterize of a speaker's voice. The codebook is generated by a clustering procedure based upon a predefined objective distortion measure, which computes the dissimilarity between any two given vectors [46]. The codebook can also be considered an implicit representation of a mixture distribution used to describe the statistical properties of the source, i.e., the particular talker. In the test session, input vectors from the unknown talker are compared with the nearest codebook entry and the corresponding distortions are accumulated to form the basis for a classification decision.

7.2.3 Stochastic Models for Non-Stationary Process

In the above section, stationarity of speech is assumed and the methods for talker identification described therein does not make use of the temporal information of speech. In many applications, the temporal information is necessary and important

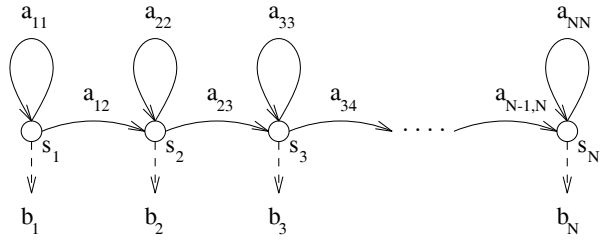

FIGURE 7.4
Left-to-right hidden Markov model.

in making a decision. A more powerful model – hidden Markov model (HMM) is then applied to characterize both the temporal structure and the corresponding statistical variations along the parameter trajectory of an utterance.

In speech and speaker recognition, an HMM is trained to represent the acoustic pattern of a subword, a word, or a whole pass-phrase. There are many variants of HMMs. The simplest kind is an N-state, left-to-right model without a state-skip as shown in Figure 7.4. This is widely used in speaker authentication. The figure shows a Markov chain with a sequence of states, representing the evolution of speech signals. Within each state, a Gaussian mixture model (GMM) is used to characterize the observed speech feature vector as a multivariate distribution.

An HMM, λ, can be completely characterized by three sets of parameters, the state transition probabilities, A, the observation densities, B, and the initial state probabilities, Π; as shown in the following notation:

$$\lambda = \{A, B, \Pi\} = \{a_{i,j}, b_i, \pi_i\}, \quad i, j = 1, ..., N, \tag{7.19}$$

where N is the total number of states. Given an observation sequence $\mathbf{O} = \{\mathbf{o}_t\}_{t=1}^{T}$, the model parameters, $\{A, B, \Pi\}$, of λ can be trained by an iterative method to optimize a prescribed performance criterion, e.g., maximum likelihood. In practice, the *segmental K-mean* algorithm [36] has been widely used. Following model initialization, the observation sequence is segmented into states based on the current model λ. Then, within each state, a new GMM is trained by the above EM algorithm to maximize the likelihood. The new HMM $\hat{\lambda}$ is then used to re-segment the observation sequence and re-estimation of model parameters ensues. The iterative procedure usually converges in a few iterations.

Other than the maximum likelihood criterion, the model can also be trained by optimizing a discriminative function. For example, the Minimum Classification Error (MCE) criterion [16] was proposed along with a corresponding generalized probabilistic descent (GPD) training algorithm [7, 15] to minimize an objective function that approximates the error rate closely. Other criteria like Maximum Mutual Information (MMI) [4, 33] have also been attempted. Instead of modeling the distribution of the data set of the target class, the criteria also incorporate data of other classes. A discriminative model is thus constructed to implicitly model the underlying distribution of the target class but with explicit emphasis on minimizing the

classification error or maximizing the mutual information between the target class and others. The discriminative training algorithms have been applied successfully to speech recognition. The MCE/GPD algorithm has also been applied to speaker recognition [29, 18, 43, 45]. Generally speaking, the models trained by discriminative objective functions yield better recognition and verification performance, but the long training time makes it less attractive to real applications.

7.2.4 Speech Segmentation

Given an HMM, λ, and a sequence of observations, $\mathbf{O} = \{\mathbf{o}_t\}_{t=1}^{T}$, the optimal state segmentation can be determined by evaluating the maximum joint state-observation probability, $\max_s P(\mathbf{O}, s|\lambda)$, conventionally called maximum likelihood decoding. One popular algorithm that accomplish this objective efficiently is the Viterbi algorithm [49, 10]. When fast decoding and forced alignment is desired, a new reduced space search algorithm [20] can be employed.

7.2.5 Statistical Verification

Statistical verification as applied to speaker verification and utterance verification can be considered as a two-class classification problem: whether a spoken utterance is from the true speaker (the target source) or from an impostor (the alternative source). Given an observation $\mathbf{o}$, a decision α_i is taken based on the following conditional risks derived from Eq. (7.1):

$$R(\alpha_1|\mathbf{o}) = \mathcal{L}(\alpha_1|C_1)P(C_1|\mathbf{o}) + \mathcal{L}(\alpha_1|C_2)P(C_2|\mathbf{o}) \tag{7.20}$$

$$R(\alpha_2|\mathbf{o}) = \mathcal{L}(\alpha_2|C_1)P(C_1|\mathbf{o}) + \mathcal{L}(\alpha_2|C_2)P(C_2|\mathbf{o}). \tag{7.21}$$

The action α_1 corresponds to the decision of positive verification if

$$R(\alpha_1|\mathbf{o}) < R(\alpha_2|\mathbf{o}). \tag{7.22}$$

Bring (7.20) and (7.21) into (7.22) and rearranging the terms, we take action α_1 if:

$$\frac{P(C_1|\mathbf{o})}{P(C_2|\mathbf{o})} > \frac{\mathcal{L}(\alpha_1|C_2) - \mathcal{L}(\alpha_2|C_2)}{\mathcal{L}(\alpha_2|C_1) - \mathcal{L}(\alpha_1|C_1)} = \mathcal{T}_1 \tag{7.23}$$

where $\mathcal{T}_1 > 1$ is a prescribed threshold. Furthermore, by applying the Bayes formula, we have

$$\frac{p(\mathbf{o}|C_1)}{p(\mathbf{o}|C_2)} > \mathcal{T}_1 \frac{P(C_2)}{P(C_1)} = \mathcal{T}_2. \tag{7.24}$$

For a sequence of observation $\mathbf{O} = \{\mathbf{o}_i\}_{i=1}^{T}$ which are assumed to be independent and identically distributed (i.i.d.), we have the likelihood-ratio test:

$$r(\mathbf{O}) = \frac{\prod_{t=1}^{T} p(\mathbf{o}_t|C_1)}{\prod_{t=1}^{T} p(\mathbf{o}_t|C_2)} = \frac{P(\mathbf{O}|C_1)}{P(\mathbf{O}|C_2)} > \mathcal{T}_3. \tag{7.25}$$

The same result can also be derived from the Neymann-Pearson decision formulation, thus the name *Neymann-Pearson* test [31, 32, 50]. It can be shown that the likelihood-ratio test minimizes the verification error for one class while maintaining the verification error for the other class constant [11, 32].
In practice, we compute a log-likelihood ratio for verification:

$$\mathcal{R}(\mathbf{O}) = \log P(\mathbf{O}|C_1) - \log P(\mathbf{O}|C_2). \tag{7.26}$$

A decision is made according to the rule:

$$\begin{cases} \text{Acceptance: } \mathcal{R}(\mathbf{O}) \geq \mathcal{T} \\ \text{Rejection:} \quad \mathcal{R}(\mathbf{O}) < \mathcal{T}, \end{cases} \tag{7.27}$$

where $\mathcal{T}$ is a threshold value, which can be determined theoretically or experimentally.
There are two types of error in a test: false rejection, i.e., rejecting the hypothesis when it is actually true, and false acceptance, i.e., accepting it when it is actually false. The equal error rate (EER) is defined as the error rate when the operating point is so chosen as to achieve equal error probabilities for the two types of error. EER has been widely used as a verification performance indicator.
In utterance verification, we assume that the expected word or subword sequence is known and the task is to verify whether the input spoken utterance matches it. Similarly, in SV, the text of the pass-phrase is known. The task is to verify whether the input spoken utterance matches the given sequence, using the model trained by the speaker's voice.

7.3 Speaker Verification System

Among different speaker verification systems introduced in Section 7.1.1, we focus here on the fixed-phrase system [34, 25] and evaluate the system in an open-set test. This is due to three reasons. First, a short, user-selected phrase is easy to remember. Second, a fixed-phrase system usually has a better performance than a text-prompted system [26]. Last, an open-set evaluation is more appropriate for real applications. For example, a large-scale tele-banking system usually involves a large user population. The population also changes on a daily basis. It is impossible and unrealistic to consider SV as a close-set problem.
As shown in Fig. 7.2, a fixed-phrase system has two phases, enrollment and test. For feature extraction, the speech signal is sampled at 8 kHz and pre-emphasized using a first-order filter with a coefficient of 0.97. The samples are blocked into overlapping frames of 30 ms in duration and updated at 10 ms intervals. Each frame is windowed with a Hamming window followed by a 10-th order LPC analysis. The LPC coefficients are then converted to cepstral coefficients, where only the first 12 coefficients

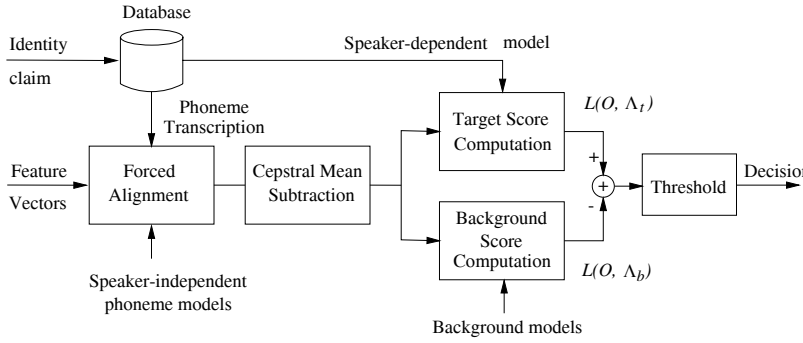

FIGURE 7.5
A fixed-phrase speaker verification system.

are retained for computing the feature vector. The feature vector consisted of 24 features including the 12 cepstral coefficients and 12 delta cepstral coefficients [34]. During enrollment, LPC cepstral feature vectors corresponding to the non-silence portion of the enrollment pass-phrases are used to train a SD left-to-right HMM to represent the voice pattern. It is called a *whole-word or whole phrase model* [34]. In addition to model training, the text of the pass-phrase collected from the enrollment session is transcribed into a sequence of phonemes, $\{S_k\}_{k=1}^{K}$, where S_k is the kth phoneme, and K is the total number of phonemes in the pass-phrase. The models and the transcription are saved in the database.

A detailed block diagram of a test session is shown in Fig. 7.5. After a speaker claims his or her identity, the system expects the user to speak the same phrase as in the enrollment session. The voice waveform is first converted to the prescribed feature representation. In the forced alignment block, a sequence of speaker-independent phoneme models is constructed according to the phonemic transcription of the pass-phrase. The model sequence is then used to segment and align the feature vector sequence through use of the Viterbi algorithm. In the cepstral mean subtraction block, silence frames are removed, and a mean vector is computed based on the remaining speech frames. The mean is then subtracted from all speech frames [12]. This is an important step for channel compensation. It makes the system more robust to changes in the operating environment as well as in the transmission channel. We note that the forced alignment block is also used for accurate endpoint detection. For a system with limited computing power, a separate endpoint detection algorithm can be implemented for fast response [27, 28].

In the block of target score computation of Fig. 7.5, speech feature vectors are decoded into states by the Viterbi algorithm, using the trained whole-phrase model. A log-likelihood score for the target model, i.e. the target score, is calculated as

$$L(\mathbf{O}, \Lambda_t) = \frac{1}{N_f} \log P(\mathbf{O}|\Lambda_t), \tag{7.28}$$

where $\mathbf{O}$ is the feature vector sequence, N_f is the total number of vectors in the

sequence, Λ_t is the target model, and $P(\mathbf{O}|\Lambda_t)$ is the likelihood score resulted from Viterbi decoding.

In the block of background (non-target) score computation, a set of speaker independent HMMs in the order of the transcribed phoneme sequence, $\Lambda_b = \{\lambda_1, ..., \lambda_K\}$, is applied to align the input utterance with the expected transcription using the Viterbi decoding algorithm. The segmented utterance is $\mathbf{O} = \{\mathbf{O}_1, ..., \mathbf{O}_K\}$, where $\mathbf{O}_i$ is the set of feature vectors corresponding to the i'th phoneme, S_i, in the phoneme sequence. The background or non-target likelihood score is then computed by

$$L(\mathbf{O}, \Lambda_b) = \frac{1}{N_f} \sum_{i=1}^{K} \log P(\mathbf{O}_i|\lambda_i), \tag{7.29}$$

where $\Lambda_b = \{\lambda_i\}_{i=1}^K$ is the set of SI phoneme models in the order of the transcribed phoneme sequence, $P(\mathbf{O}_i|\lambda_{b_i})$ is the corresponding phoneme likelihood score, and K is the total number of phonemes.

The target and background scores [42] are then used in the likelihood-ratio test:

$$\mathcal{R}(\mathbf{O}; \Lambda_t, \Lambda_b) = L(\mathbf{O}, \Lambda_t) - L(\mathbf{O}, \Lambda_b), \tag{7.30}$$

where $L(\mathbf{O}, \Lambda_t)$ and $L(\mathbf{O}, \Lambda_b)$ are defined in Eqs. (7.28) and (7.29) respectively.

The system has been tested on a database consisting of fixed-phrase utterances. The database was recorded over a long-distance telephone network. It consists of 100 speakers, 51 male and 49 female. The fixed phrase, common to all speakers, is "I pledge allegiance to the flag" with an average utterance length of 2 seconds. Five utterances from each speaker recorded in one enrollment session (one telephone call) are used to construct an SD target HMM. For testing, we used 50 utterances recorded from a true speaker in different sessions (from different telephone channels and handsets at different time with different background noise), and 200 utterances recorded from 51 or 49 impostors of the same gender in different sessions [34, 25].

The SD target models for the phrases are left-to-right HMMs. The number of states depends on the total number of phonemes in the phrases. There are 4 Gaussian mixture components associated with each state [34]. The background models are concatenated SI phone HMMs trained on a telephone speech database from different speakers and text [42]. There are 43 HMMs, corresponding to 43 phonemes respectively, and each model has three states with 32 Gaussian components per state. Again, due to unreliable variance estimates from a limited amount of speaker-specific training data, a global variance estimate was used as the common variance to all Gaussian components in the target models [34].

In order to further improve the SD HMM, a model adaptation procedure is employed. The second, fourth, sixth, and eighth test utterances from the true speaker, which were recorded at different times, are used to update the means and mixture weights of the SD HMM for verifying successive test utterances. For the above database, the average individual equal-error rate over 100 speakers is 2.6% without adaptation and 1.8% with adaptation, respectively [25], as shown in Table 7.1. In general, the longer the pass-phrase, the higher the accuracy. The response time depends on the hardware/software configuration. For most cases, a real time response is expected.

TABLE 7.1
Experimental Results in Average Equal-Error Rates

	Without Adaptation	With Adaptation
Fixed Pass-Phrase Speaker Verification	2.61%	1.80%

(Tested on 100 speakers using one common pass-phrase.)

We note that the same pass-phrase is used for all speakers in our evaluation, and the above results are the lower bound of the performance. The actual system performance would be better when users choose their own and most likely different pass-phrase. Also, to ensure the open test nature, none of the impostor's data was used for discriminatively training the SD target model.

7.4 Verbal Information Verification

In this section, we introduce a pattern recognition technique for verbal information verification (VIV) and present some experimental results [24].

Generally speaking, there are two ways to verify a single spoken utterance for VIV: by automatic speech recognition (ASR) or by utterance verification (UV). With ASR, the spoken input is transcribed into a sequence of words. The transcribed words are then compared to the information pre-stored in the claimed speaker's personal profile. With UV, the spoken input is verified against an expected sequence of words or subwords, which is taken from a personal data profile of the claimed individual. Based on our experience [23] and the analysis, the utterance verification approach can give us much better performance than the ASR approach. Therefore, we focus on the utterance verification approach in this study.

When a question is answered in the form of a naturally spoken utterance, the key information in the profile may be embedded in a sentence, e.g. "My mother's maiden name is ..." In the sentence, the name is the key information, which can be extracted with a keyword spotting technique [51]. Here, we assume that the key information has been extracted or the answered utterance contains only the key information.

To verify one single utterance, we employ the technique of utterance verification, which was developed for keyword sporting and non-keyword rejection (e.g. [37, 38, 17, 30, 47, 48, 44]). A block diagram of the utterance verification for VIV is shown in Fig. 7.6. The three key modules are: utterance segmentation by forced decoding, subword testing, and utterance level confidence measure calculation. They will be described in detail in the following subsections.

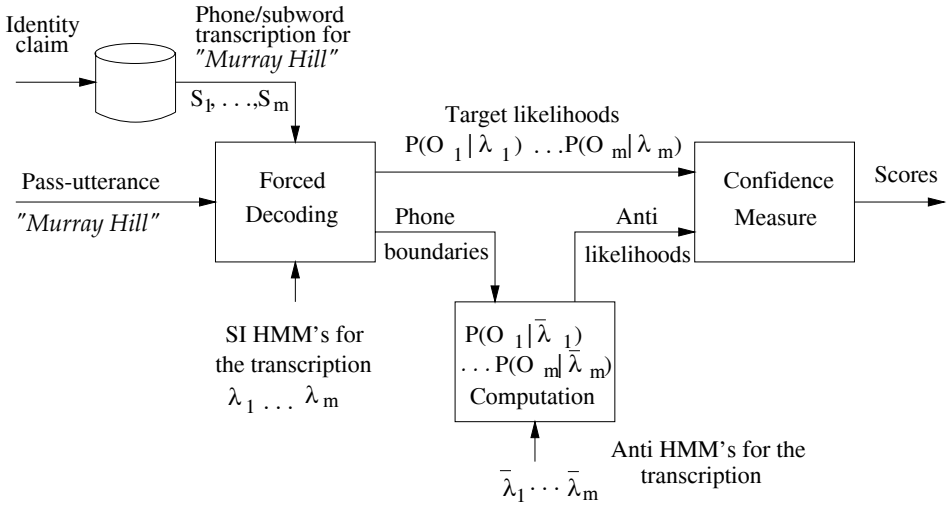

FIGURE 7.6
Utterance verification in verbal information verification (VIV).

7.4.1 Utterance Segmentation

When a user opens an account, key information that constitutes his or her profile is registered in a database. Each piece of the key information is represented by a sequence of words, $\mathbf{S}$, which in turn is equivalently characterized by a concatenation of a sequence of phonemes or subwords, $\{S_n\}_{n=1}^{N}$, where S_n is the n-th subword, and N is the total number of subwords in the key word sequence.
Since the VIV system only prompts one single question at a time, the system knows the expected (correct) key information to the prompted question and the corresponding subword sequence $\mathbf{S}$. We then apply the subword models $\lambda_1, ..., \lambda_N$ in the same order of the subword sequence $\mathbf{S}$ to decode the answer utterance using the Viterbi algorithm introduced previously. This can be represented as:

$$P(\mathbf{O}|\mathbf{S}) = \max_{T_1,T_2,...,T_N} P(O_1^{T_1}|S_1)P(O_{T_1+1}^{T_2}|S_2)\ldots P(O_{T_{N-1}+1}^{T_N}|S_N), \qquad (7.31)$$

where

$$\mathbf{O} = \{\mathbf{O}_1, \mathbf{O}_2, ..., \mathbf{O}_N\} = \{O_1^{T_1}, O_{T_1+1}^{T_2}, ..., O_{T_{N-1}+1}^{T_N}\}, \qquad (7.32)$$

is a set of segmented feature vectors associated with subwords, $T_1, T_2, ..., T_N$ are the end-frame numbers of each subword segments, respectively, and $\mathbf{O}_n = O_{T_{n-1}+1}^{T_n}$ is the segmented sequence of observations corresponding to subword S_n, from frame number $T_{n-1}+1$ to frame number T_n, where $T_1 \geq 1$ and $T_i > T_{i-1}$.

7.4.2 Subword Hypothesis Testing

Given a decoded subword, S_n, in an observed speech segment $\mathbf{O}_n$, we need a decision rule by which we assign the subword to either one of the two classes: hypotheses

H_0 or H_1, where H_0 means that the observed speech $\mathbf{O}_n$ consists of the actual sound of subword S_n, and H_1 is the alternative hypothesis. The most powerful test is the likelihood-ratio test as we have introduced:

$$r(\mathbf{O}_n) = \frac{P(\mathbf{O}_n|H_0)}{P(\mathbf{O}_n|H_1)} = \frac{P(\mathbf{O}_n|\lambda_n)}{P(\mathbf{O}_n|\bar{\lambda}_n)}, \tag{7.33}$$

where λ_n and $\bar{\lambda}_n$ are the target HMM and corresponding anti-HMMs for subword unit S_n respectively. The target model, λ_n, is trained using the data of subword S_n; the corresponding anti-model, $\bar{\lambda}_n$ is trained using the data of a set of subwords $\bar{\mathbf{S}}$ which is highly confusable with subword S_n [47], i.e. $\bar{\mathbf{S}}_n \subset \{S_i\}, i \neq n$. The log likelihood ratio (LLR) for subword S_n is

$$R(\mathbf{O}_n) = \log P(\mathbf{O}_n|\lambda_n) - \log P(\mathbf{O}_n|\bar{\lambda}_n). \tag{7.34}$$

For normalization, an average frame LLR, R_n, is defined as

$$R_n = \frac{1}{T_n}\left[\log P(\mathbf{O}_n|\lambda_n) - \log P(\mathbf{O}_n|\bar{\lambda}_n)\right], \tag{7.35}$$

where T_n is the length of the speech segment. For each subword, a decision can be made by

$$\begin{cases} \text{Acceptance: } R_n \geq \mathcal{T}_n; \\ \text{Rejection: } \quad R_n < \mathcal{T}_n, \end{cases} \tag{7.36}$$

where either a subword-dependent threshold value $\mathcal{T}_n$ or a common threshold $\mathcal{T}$ can be determined numerically or experimentally.

7.4.3 Confidence Measure Calculation

For an utterance level decision, we have to define a function to combine the results of subword tests. A confidence measure $\mathcal{M}$ for a key utterance $\mathbf{O}$ can be represented as

$$\mathcal{M}(\mathbf{O}) = \mathcal{F}(R_1, R_2, ..., R_N), \tag{7.37}$$

where $\mathcal{F}$ is the function to combine the LLRs of all subwords in the key utterance. Several confidence measures have been proposed for utterance verification [17, 30]. We denote two of them as M_1 and M_2 in the following.

$$M_1 = \frac{1}{L}\sum_{n=1}^{N} l_n R_n, \tag{7.38}$$

where N is the total number of non-silence subwords in the utterance, and L is the total number of frames of the non-silent portion of the utterance, i.e. $L = \sum_{n=1}^{N} l_n$. Furthermore,

$$M_2 = \frac{1}{N}\sum_{n=1}^{N} R_n. \tag{7.39}$$

Here, M_1 is an average score over all frames and all subwords. Each of the subword score R_n is weighted by its duration. M_2 is an average LLR of all subwords and independent of individual durations. We note that silence models are used during forced alignment for utterance segmentation but only non-silence subwords are involved in computing the confidence measures.

For VIV, we defined a different confidence measure, M, for two reasons. First, as reported in [17] and from our experiments, the above confidence measures have a large dynamic range. A preferable statistic should have a stable, limited numerical range, such that a common threshold can be determined for all subwords to simplify the operation. Second, decision thresholds should be determined to meet specifications in different applications. It is desirable to be able to relate the design specifications with the computed confidence measure.

A useful design specification is the percentage of acceptable subwords in a key utterance. We then need to make a decision at both the subword and the utterance level. At the subword level, a likelihood-ratio test can be conducted to reach a decision to accept or reject each subword. At the utterance level, a simple utterance score can be computed to represent the percentage of acceptable subwords.

To make a decision at the subword level, we need to determine the threshold for each of the subword tests. If we have the training data for each subword model and the corresponding anti-subword model, this is not a problem. However, in many cases, the data may not be available. Therefore, we need to define a test that can conveniently determine the thresholds without using the training data. For subword S_n which is characterized by a model, λ_n, we define

$$C_n = \frac{\log P(\mathbf{O}_n|\lambda_n) - \log P(\mathbf{O}_n|\bar{\lambda}_n)}{\log P(\mathbf{O}_n|\lambda_n)} \tag{7.40}$$

where $\log P(\mathbf{O}_n|\lambda_n) \neq 0$. $C_n > 0$ means the target score is larger than the anti-score and vice versa. Furthermore, we define a *normalized confidence measure* for an utterance with N subwords as

$$M = \frac{1}{N}\sum_{n=1}^{N} f(C_n), \tag{7.41}$$

where

$$f(C_n) = \begin{cases} 1, \text{ if } C_n \geq \theta; \\ 0, \text{ otherwise,} \end{cases} \tag{7.42}$$

M is in a fixed range of $0 \leq M \leq 1$. Due to the normalization in Eq. (7.40), θ is a subword-independent threshold which can be determined separately. A subword is accepted and counted as part of the utterance confidence measure only if its C_n score is greater than or equal to the threshold value θ. Thus, M can be interpreted as the percentage of acceptable subwords in an utterance; e.g. $M = 0.8$ implies that 80% of the subwords in the utterance are acceptable. Therefore, an utterance threshold can be determined or adjusted based on the specifications of system performance and robustness.

7.4.4 Sequential Utterance Verification

For VIV, the system would go through more than one question-answer turns before a final decision is made. Thus, the above single utterance test strategy needs to be extended to a sequence of subtests, similar to the *step-down procedure* in statistics [1]. In such a sequential test, each of the subtests is independently constructed as a single-utterance verification test. We can make a soft or delayed decision for test i as follows:

$$\begin{cases} \text{Acceptance:} & M(i) > \mathcal{T}_H(i); \\ \text{Delay (go to the next test):} & \mathcal{T}_L(i) \leq M(i) \leq \mathcal{T}_H(i); \\ \text{Rejection:} & \mathcal{T}_L(i) > M(i), \end{cases} \tag{7.43}$$

where $M(i)$ is a confidence score, $\mathcal{T}_H(i)$ and $\mathcal{T}_L(i)$ are the high and low thresholds for test i, respectively.
Let $\mathcal{H}_0$ be the target hypothesis in which all the answered utterances match the key information in the profile. We have

$$\mathcal{H}_0 = \bigcap_{i=1}^{J} H_0(i), \tag{7.44}$$

where J is the total number of subtests, and $H_0(i)$ is a component target hypothesis in the i-th subtest corresponding to the i-th utterance. The alternative hypothesis is

$$\mathcal{H}_1 = \bigcup_{i=1}^{J} H_1(i), \tag{7.45}$$

where $H_1(i)$ is a component alternative hypothesis corresponding to the i-th subtest. On the i-th subtest, a simplified version of the soft decision can be made as:

$$\begin{cases} \text{Delay (go to the next test):} & M(i) \geq \mathcal{T}(i); \\ \text{Rejection:} & M(i) < \mathcal{T}(i), \end{cases} \tag{7.46}$$

where $M(i)$ is a confidence score, and $\mathcal{T}(i)$ is a single thresholds for test i.
As we have introduced, when performing a hypothesis test, one may commit one of two types of errors: rejecting the hypothesis when it is true - *false rejection* (FR), or accepting it when it is false - *false acceptance* (FA). We denote the FR and FA error rates as ε_r and ε_a, respectively. An *equal-error rate* (EER), ε, is defined when the two error rates are made equal by choosing a particular operating point for the system, i.e. $\varepsilon_r = \varepsilon_a = \varepsilon$. For a sequential test, we extend the definitions of error rates as follows.
Definition 1: *False rejection error on J utterances* ($J \geq 1$) is the error when the system rejects a correct response in any one of J hypothesis subtests.
Definition 2: *False acceptance error on J utterances* ($J \geq 1$) is the error when the system accepts an incorrect set of responses after all of J hypothesis subtests.
Definition 3: *Equal-error rate on J utterances* is the rate at which the false rejection error rate and the false acceptance error rate on J utterances are equal.

We denote the above FR and FA error rates on J utterances as $E_r(J)$ and $E_a(J)$, respectively. Let $\Omega_i = \mathcal{R}_1(i) \cup \mathcal{R}_0(i)$ be the region of confidence scores of the i-th subtest, where $\mathcal{R}_0(i)$ is the region of confidence scores which satisfy $M(i) \geq T(i)$ from which we accept $H_0(i)$, and $\mathcal{R}_1(i)$ is the region of scores which satisfy $M(i) < T(i)$ from which we accept $H_1(i)$.

The FR and FA errors for subtest i can be represented as the following conditional probabilities

$$\varepsilon_r(i) = P(\, M(i) \in \mathcal{R}_1(i) \mid H_0(i) \,), \tag{7.47}$$

and

$$\varepsilon_a(i) = P(\, M(i) \in \mathcal{R}_0(i) \mid H_1(i) \,), \tag{7.48}$$

respectively. Furthermore, the FR error on J utterances can be evaluated as

$$\begin{aligned} E_r(J) &= P(\, \bigcup_{i=1}^{J} \{M(i) \in \mathcal{R}_1(i)\} \mid \mathcal{H}_0 \,), \\ &= 1 - \prod_{i=1}^{J} (1 - \varepsilon_r(i)), \end{aligned} \tag{7.49}$$

and the FA error on J utterances is

$$\begin{aligned} E_a(J) &= P(\, \bigcap_{i=1}^{J} \{M(i) \in \mathcal{R}_0(i)\} \mid \mathcal{H}_1 \,), \\ &= \prod_{i=1}^{J} \varepsilon_a(i). \end{aligned} \tag{7.50}$$

Eqs. (7.49) and (7.50) indicate an important property of the sequential test defined above: the more the subtests, the less the FA error and the larger the FR error. Therefore, it is important that the threshold at every subtest is carefully chosen so as to achieve an FR error, ε_r, that is close to zero or a small number corresponding to the design specification, but add more subtests in the same way as needed until the required system FA error rate, E_a is met, or the maximum number of allowed subtests is reached.

It is reasonable to arrange the subtests in the order of descending importance and/or decreasing subtest error rates. In other words, the system first prompts users with the most important question or with the subtest that we know has the lowest FR error $\varepsilon_r(i)$. Therefore, if a speaker is falsely rejected, the session can be restarted right away with little inconvenience to the user.

Eq. (7.49) also indicates the reason that the ASR approach would not perform very well in a sequential test. Although an ASR can achieve low FR error, $\varepsilon_r(i)$, on each of the individual subtests, the overall FR error on J utterances $E_r(J)$, $J > 1$, can still be very high due to the fact that the verification process in an ASR-based approach uses word comparison and does not permit a soft decision or delayed decision. In the proposed utterance verification approach, the FR on each individual subtest is

made close to zero by adjusting the threshold value while controlling the overall FA error by adding more subtests until reaching the design specification. We use the following examples to show the above concept.

Example 1: A bank operator usually asks two kinds of personal questions when verifying a customer. When automatic VIV is applied to the procedure, the average individual error rates on these two subtests are $\varepsilon_r(1) = 0.1\%$, $\varepsilon_a(1) = 5\%$; and $\varepsilon_r(2) = 0.2\%$, $\varepsilon_a(2) = 6\%$, respectively. Then, from Eq. (7.49) and (7.50), we know that the system FR and FA errors on a sequential test are $E_r(2) = 0.3\%$ and $E_a(2) = 0.3\%$. If the bank wants to further reduce the FA error, one additional subtest can be added to the sequential test. Suppose the additional subtest has $\varepsilon_r(3) = 0.3\%$ and $\varepsilon_a(3) = 7\%$. The overall system error rates will be $E_r(3) = 0.6\%$ and $E_a(3) = 0.021\%$.

Example 2: A security system requires $E_r(J) \leq 0.03\%$ and $E_a(J) \leq 0.2\%$. It is known that each subtest can have $\varepsilon_r \leq 0.01\%$, and $\varepsilon_a \leq 12\%$ by adjusting the thresholds. In this case, we need to determine the number of subtests, J, to meet the design specifications. From Eq. (7.50), we have

$$J = \left\lceil \frac{\log E_a}{\log \varepsilon_a} \right\rceil = \left\lceil \frac{\log 0.002}{\log 0.12} \right\rceil = 3.$$

Then, the actual system FA rate on three subtests is $E_a(3) = 0.17\% \leq 0.2\%$; the FR rate on three tests is $E_a(3) = 0.03\%$. Therefore, three subtests can meet the required performance on both FR and FA.

7.4.5 VIV Experimental Results

In the following experiments, the VIV system verifies speakers by three sequential subtests, i.e. $J = 3$. The experimental database includes 100 speakers. Each speaker provided three utterances as the answers to the following three questions:

"In which year were you born?"

"In which city and state did you grow up?" and

"May I have your telephone number, please?"

The database we used is a biased one. Twenty six percent (26%) of the speakers have birth year in the 1950s and 24% are in the 1960s. There is only one digit that differentiates those birth years. In city and state names, 39% are "New Jersey", and 5% of the speakers used exactly the same answer, "Murray Hill, New Jersey". Thirty eight percent (38%) of the telephone numbers start from "908 582 ...", which means that at least 60% of the digits in their answer for the telephone number are identical. Also, some of the speakers have foreign accent, and some cities and states are in foreign countries. Existing ASR systems cannot provide an acceptable performance. In this experiment, a speaker is considered a true speaker when the speaker's utterances are verified against his or her data profile. The same speaker is used as an impostor when the utterances are verified against other speakers' profiles. Thus, for each true speaker, we have three utterances from the speaker and 99 $\times$ 3 utterances from other 99 speakers as impostors.

The feature vector consisted of 39 features including 12 LPC cepstral coefficients, 12 delta cepstral coefficients, 12 delta-delta cepstral coefficients, energy, delta energy, and delta-delta energy [35]. In evaluating the subword verification scores a set of 1117 right context-dependent HMMs were used as the target phone models [19], and a set of 41 context-independent anti-phone HMMs as anti-models [47].

For a VIV system with multiple subtests, one can use either one single threshold applied globally to all the subtests, i.e. $\mathcal{T} = \mathcal{T}(i)$, or multiple thresholds, each applied to individual questions respectively, i.e. $\mathcal{T}(i) \neq \mathcal{T}(j)$, $i \neq j$. The thresholds can be either context dependent or context independent. They can also be either speaker dependent or speaker independent. A VIV system can start from a speaker-independent threshold, then switch to speaker- and context-dependent thresholds after the system has been used for several times by a user. To ensure no false rejection, the upper bound of the threshold for subtest i of a speaker can be selected as

$$T(i) \leq \min\{M(i,j)\}, \qquad j = 1, ..., I, \tag{7.51}$$

where $M(i,j)$ is the confidence score for utterance i on the jth trial, and I is the total number of trials that the speaker has performed in the same context of utterance i. The thresholds can also be updated based on the recent scores to accommodate the changes in speaker's voice and environment.

In this experiment, we used three thresholds associated with the three questions for each speaker. Following the design strategy proposed in Section 7.4.4, the thresholds were determined by estimating $T(i)$ as in Eq. (7.51) to guarantee 0% false rejection rate.

TABLE 7.2
Summary of the Experimental Results on Verbal Information Verification

Approaches	False Rejection	False Acceptance	Accuracy
Sequential Utterance Verification	0%	0%	100%

(Tested on 100 speakers with 3 questions while speaker-dependent thresholds were applied.)

A summary of VIV for speaker authentication is shown in Table 7.2. When SD thresholds are set for each key information field, we achieved 0% individual equal-error rate on average. The robustness of the system was also evaluated. Interested readers are referred to [24] for details.

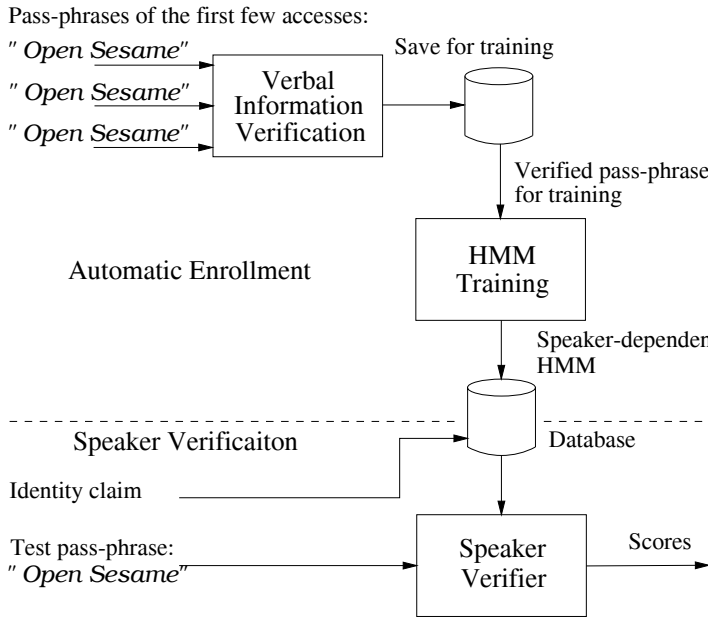

FIGURE 7.7
An integrated voice authentication system combining verbal information verification and speaker verification.

7.5 Speaker Authentication by Combining SV and VIV

In the above sections, we have introduced SV and VIV as two independent authentication techniques. In this section, we combine them together to construct a new speaker authentication system, which is more convenient to users and provides better authentication performances. Actually, these two techniques can be combined in various ways for different applications [24, 21].

As introduced above, a conventional SV system as shown in Fig 7.2 involves two kinds of sessions: enrollment and testing. In the enrollment session, the system asks the user to utter their pass-phrase several times to allow training of speaker-dependent models. In real applications, we found that users often make mistakes during enrollment. This kind of error is very difficult to correct once a speaker-dependent model is constructed, unless manual examination and verification of data takes place before model training. Obviously, VIV is a natural and powerful technique for this purpose. One of the solutions is shown in Fig. 7.7. During the first few accesses or uses of the system, authentication is conducted by a VIV procedure. The uttered pass-phrase must pass VIV tests; otherwise, the user is prompted to repeat. Verified utterances of the pass-phrase are then saved, and used to train a

speaker-dependent HMM for SV. At this point, the authentication system can then be switched from VIV to SV.

There are several advantages by using the combined system. First, the system is convenient to users since it does not need a formal enrollment session and a user can start to use the system right after his/her account is set up. Second, the acoustic mismatch problem is to a certain degree mitigated since the training data may come from different sessions, potentially via different handsets and channels. Third, the quality of the training data are ensured since the training phrases are verified by VIV before being used to train the speaker-dependent HMMs for the pass-phrase. Finally, once the system switches to SV, it would be difficult for an impostor to access the account even if the imposter knows the true speaker's pass-phrase.

We conducted an experiment to verify the performance of the combined system. The feature and database are the same as the speaker verification system introduced in the previous section. The experimental database consists of fixed phrase utterances recorded over the long distance telephone network by 100 speakers, 51 male and 49 female. The fixed phrase, common to all speakers, is "I pledge allegiance to the flag" with an average length of 2 seconds. We assume the fixed phrase is one of the verified utterances in VIV. Five utterances of the pass-phrase recorded from five separate VIV sessions were used to train the SD HMM, thus the training data are collected from different acoustic environments and telephone channels at different time. We assume all the collected utterances have been verified by VIV to ensure the quality of the training data.

For testing, we used 40 utterances recorded from a true speaker in different sessions, and 192 utterances recorded from 50 impostors of the same gender in different sessions. The model structure is the same as the previous SV system. For model adaptation, the second, fourth, sixth, and eighth test utterances from the tested true speaker were used to update the associated HMMs for verifying subsequent test utterances incrementally [34].

In Section 7.4.5, we have reported the experimental results of VIV in a test of 100 speakers. The system achieved 0% error rates with three rounds of question-answer test in a sequential utterance verification procedure. Therefore, we assume that all the training utterances collected by VIV are correct. In other words, while improvement by reducing acoustic mismatch will become obvious in the result, we did not design an experiment to show the potential improvement in verification performance from an increased sanity check on the training data.

The SV experimental results without and with adaptation are listed in Table 7.3 and Table 7.4 for the 100 speakers, respectively. The numbers are expressed in terms of the average percentage of individual equal-error rate (EER). The first data column lists the EERs using individual thresholds, and the second data column lists the EERs using common (pooled) thresholds for all tested speakers.

The baseline system is the conventional SV system in which a single enrollment session is used. In the combined system, VIV is used for the automatic enrollment for SV. After the VIV system is used for five times that allows collection of training utterances from five different sessions, it switches to the SV procedure. The test utterances for both the baseline and the proposed system are the same.

Without adaptation, the baseline system has an EER of 3.03% and 4.96% for individual and pooled thresholds, respectively, while the proposed system has an EER of 1.59% and 2.89%, respectively. With adaptation as defined in the last subsection, the baseline system achieves an EER of 2.15% and 3.12%, while the proposed system achieves an EER of 1.20% and 1.83%, respectively. The proposed system without adaptation has an even lower EER than the baseline system with adaptation. This is because the SD models in the proposed system were trained using the data from different sessions while the baseline system just performed an incremental adaptation without reconstructing the models after collecting more data.

TABLE 7.3
Experimental Results without Adaptation in Average Equal-Error Rates

Algorithms	Individual Thresholds	Pooled Thresholds
SV (Baseline)	3.03 %	4.96 %
VIV+SV(proposed)	1.59 %	2.89 %

TABLE 7.4
Experimental Results with Adaptation in Average Equal-Error Rates

Algorithms	Individual Thresholds	Pooled Thresholds
SV (Baseline)	2.15 %	3.12 %
VIV+SV(proposed)	1.20 %	1.83 %

The experimental results indicate several advantages of the proposed system. First, since VIV can provide the training data from different sessions representing different channel environments, the system can perform significantly better than one with single-session training. Second, although it is possible to adapt the models originally trained with the single-session data to new test environments, the combined system appears to perform better still. This is due to the fact that a new model constructed with multi-session training data is more accurate than that with incremental adaptation using the multi-session data. Lastly, in real-world applications, all the utterances used in training and adaptation can be verified by VIV before training or adaptation. Although this advantage cannot be observed in this database evaluation, it is critical in real-world applications since even a true speaker may make a mistake while uttering a pass-phrase. The mistake will never be corrected once involved in model training or adaptation. VIV can protect the system from wrong training data.
In this section, we only proposed one configuration of a combined authentication system. For different applications, different configurations of integration can be designed to meet the specification. Finally, we note that it is the user's responsibility to protect his or her personal information from impostors until the SD model is trained

and the system is migrated to an SV system. After migration, an impostor would have difficulties in accessing the account even if the pass-phrase is known.

7.6 Summary

In this chapter, we presented pattern recognition methods in speaker authentication. The theoretical foundation of the authentication techniques is the Bayesian decision theory and hypothesis testing. Depending on applications, hypothesis testing can be conducted at phrase, word, phoneme, or subword level. One extension to the Bayesian theory to authentication is the sequential verification procedure. Given a number of test utterances (subtests), the test procedure can be designed to achieve minimal overall error rate. The sequential verification procedure can also be applied to speaker verification to reduce the error rate.

Among the authentication techniques, speaker verification (SV) is the process of verifying speakers by their voice characteristics. Currently, the fixed-phrase SV system is more attractive to real applications due to its good performance. A fixed-phrase system allows users to select their personal pass-phrase; therefore, it is easy to remember and convenient to use. When an account number, such as a connected digit string, is used as a pass-phrase, the uttered account number can be recognized by an automatic speech recognition system, and then verified by a speaker verification system. Thus, one utterance can be used for both information retrieval and authentication.

Verbal information verification (SV) is to verify a speaker by the verbal content in the utterance instead of voice characteristics. We have shown that VIV can achieve very good accuracy by applying a sequential verification technique. However, since VIV is to verify the verbal content instead of the voice characteristics, it is the users' responsibility to protect their personal information from impostors.

To improve the user convenience and system performance, we further combined verbal information verification and speaker verification to construct a progressive integrated speaker authentication system. In the system, VIV is used to verify a user during the first few accesses. Simultaneously, the system collects verified training data for constructing speaker-dependent models. Later, the system migrates to an SV system for authentication. The combined system is convenient to users since they can start to use the system without going through a formal enrollment session and waiting for model training. Furthermore, since the training data may be collected from different channels in different VIV sessions, the acoustic mismatch problem is mitigated, potentially leading to a better system performance in test sessions. The SD HMMs can be updated to cover different acoustic environments while the system is in use to further improve the system performance. VIV can also be used to ensure training data for SV. For different applications, various authentication systems can be designed based on the theory and techniques presented in this chapter. A good

speaker authentication system for real applications could come from a proper integration of speaker verification, verbal information verification, speech recognition, and text-to-speech systems.

References

[1] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*, second edition. John Wiley & Sons, New York, 1984.

[2] B. S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustical Society of America*, 55:1304–1312, 1974.

[3] B. S. Atal. Automatic recognition of speakers from their voices. *Proceeding of the IEEE*, 64:460–475, 1976.

[4] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 49–52, Tokyo, 1986.

[5] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.*, 41:164–171, 1970.

[6] J. P. Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, Sept. 19.

[7] W. Chou, C.-H. Lee, and B.-H. Juang. Segmental GPD training of HMM based speech recognizer. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 473–476, San Francisco, March 1992.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39(1):1–38, 1977.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, Second Edition*. John Wiley & Sons, New York, 2001.

[10] G. D. Forney. The Viterbi algorithm. *Proceeding of IEEE*, 61:268–278, March 1973.

[11] K. Fukunaga. *Introduction to Statistical Pattern Recognition*, second edition. Academic Press, Inc., New York, 1990.

[12] S. Furui. Cepstral analysis techniques for automatic speaker verification. *IEEE Trans. Acoust., Speech, Signal Processing*, 27:254–277, April 1981.

[13] H. Gish and M. Schmidt. Text-independent speaker identification. *IEEE Signal Processing Magazine*, pages 18–32, Oct. 1994.

[14] B.-H. Juang. Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal*, 64(6):1235–1249, July-august 1985.

[15] B.-H. Juang, W. Chou, and C.-H. Lee. Minimum classification error rate methods for speech recognition. *IEEE Trans. on Speech and Audio Process.*, 5(5):257–265, May 1997.

[16] B.-H. Juang and S. Katagiri. Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*, 40(12):3043–3054, December 1992.

[17] T. Kawahara, C.-H. Lee, and B.-H. Juang. Combining key-phrase detection and subword-based verification for flexible speech understanding. In *Proceedings of ICASSP*, pages 1159–1162, Munich, May 1997.

[18] F. Korkmazskiy and B.-H. Juang. Discriminative adaptation for speaker verification. In *Proceedings of Int. Conf. on Spoken Language Processing*, volume 3, pages 28–31, Philadelphia, 1996.

[19] C.-H. Lee, B.-H. Juang, W. Chou, and J. J. Molina-Perez. A study on task-independent subword selection and modeling for speech recognition. In *Proc. of ICSLP*, pages pp. 1816–1819, Philadelphia, Oct. 1996.

[20] Q. Li. A detection approach to search-space reduction for HMM state alignment in speaker verification. *IEEE Trans. on Speech and Audio Processing*, 9(5):569–578, July 2001.

[21] Q. Li and B.-H. Juang. Speaker verification using verbal information verification for automatic enrollment. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, May 1998.

[22] Q. Li, B.-H. Juang, C.-H. Lee, Q. Zhou, and F. K. Soong. Recent advancements in automatic speaker authentication. *IEEE Robotics & Automation magazine*, 6(1):24–34, March 1999.

[23] Q. Li, B.-H. Juang, Q. Zhou, and C.-H. Lee. Verbal information verification. In *Proceedings of EUROSPEECH*, pages 839–842, Rhode, Greece, Sept. 22-25 1997.

[24] Q. Li, B.-H. Juang, Q. Zhou, and C.-H. Lee. Automatic verbal information verification for user authentication. *IEEE Trans. on Speech and Audio Processing*, 8(5):585–596, Sept. 2000.

[25] Q. Li, S. Parthasarathy, and A. E. Rosenberg. A fast algorithm for stochastic matching with application to robust speaker verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1543–1547, Munich, April 1997.

[26] Q. Li, S. Parthasarathy, A. E. Rosenberg, and D. W. Tufts. Normalized discriminant analysis with application to a hybrid speaker-verification system. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, May 1996.

[27] Q. Li and A. Tsai. A matched filter approach to endpoint detection for robust speaker verification. In *Proceedings of IEEE Workshop on Automatic Identification*, Summit, NJ, Oct. 1999.

[28] Q. Li, J. Zheng, A. Tsai, and Q. Zhou. Robust endpoint detection and energy normalization for real-time speech and speaker recognition. *IEEE Trans. on Speech and Audio Processing*, 10(3):146–157, March 2002.

[29] C. S. Liu, C.-H. Lee, W. Chou, B.-H. Juang, and A. E. Rosenberg. A study on minimum error discriminative training for speaker recognition. *Journal of the Acoustical Society of America*, 97(1):637–648, January 1995.

[30] E. Lleida and R. C. Rose. Efficient decoding and training procedures for utterance verification in continuous speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 507–510, Atlanta, May 1996.

[31] J. Neyman and E. S. Pearson. On the use and interpretation of certain test criteria for purpose of statistical inference. *Biometrika*, 20A:Pt I, 175–240; Pt II, 1928.

[32] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Phil. Trans. Roy. Soc. A*, 231:289–337, 1933.

[33] Y. Normandin, R. Cardin, and R. D. Mori. High-performance connected digit recognition using maximum mutual information estimation. *IEEE Trans. on Speech and Audio Processing*, 2(2):299–311, April 1994.

[34] S. Parthasarathy and A. E. Rosenberg. General phrase speaker verification using sub-word background models and likelihood-ratio scoring. In *Proceedings of ICSLP-96*, Philadelphia, October 1996.

[35] L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*. PTR Prentice Hall, Englewood Cliffs, NJ, 1993.

[36] L. R. Rabiner, J. G. Wilpon, and B.-H. Juang. A segmental k-means training procedure for connected word recognition. *AT&T Technical Journal*, 65(3):21–31, May/June 1986.

[37] M. G. Rahim, C.-H. Lee, and B.-H. Juang. Robust utterance verification for connected digits recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 285–288, Detroit, May 1995.

[38] M. G. Rahim, C.-H. Lee, B.-H. Juang, and W. Chou. Discriminative utterance verification using minimum string verification error (MSVE) training. In *Proc. IEEE Int. Conf. Acoustic, Speech, Signal Processing*, pages 3585–3588, Atlanta, May 1996.

[39] D. Reynolds. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Trans. on Speech and Audio Processing*, 3(1):72–83, 1993.

[40] A. E. Rosenberg. Automatic speaker verification: a review. *Proceedings of the IEEE*, 64(4):475–487, April 1976.

[41] A. E. Rosenberg and J. DeLong. HMM-based speaker verification using a telephone network database of connected digital utterances. Technical Memorandum BL01126-931206-23TM, AT&T Bell Laboratories, December 1993.

[42] A. E. Rosenberg and S. Parthasarathy. Speaker background models for connected digit password speaker verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 81–84, Atlanta, May 1996.

[43] A. E. Rosenberg, O. Siohan, and S. Parthasarathy. Speaker verification using minimum verification error training. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 105–108, Seattle, May 1998.

[44] A. R. Setlur, R. A. Sukkar, and J. Jacob. Correcting recognition errors via discriminative utterance verification. In *Proc. Int. Conf. on Spoken Language Processing*, pages 602–605, Philadelphia, Oct. 1996.

[45] O. Siohan, A. E. Rosenberg, and S. Parthasarathy. Speaker identification using minimum verification error training. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 109–112, Seattle, May 1998.

[46] F. K. Soong, A. E. Rosenberg, and B.-H. Juang. A vector quantization approach to speaker recognition. *AT&T Technical Journal*, 66:14–26, March/April 1987.

[47] R. A. Sukkar and C.-H. Lee. Vocabulary independent discriminative utterance verification for non-keyword rejection in subword based speech recognition. *IEEE Trans. Speech and Audio Process.*, 4(6):420–429, November 1996.

[48] R. A. Sukkar, A. R. Setlur, M. G. Rahim, and C.-H. Lee. Utterance verification of keyword string using word-based minimum verification error (WB-MVE) training. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 518–521, Atlanta, May 1996.

[49] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, April 1967.

[50] A. Wald. *Sequential analysis*. Chapman & Hall, NY, 1947.

[51] J. G. Wilpon, L.R. Rabiner, C.-H. Lee, and E. Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, 38(11):1870–1878, November 1990.

[52] C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

8

HMMs for Language Processing Problems

Richard M. Schwartz and John Makhoul
BBN Technologies, Verizon

CONTENTS

This chapter describes hidden Markov model (HMM) methods for various problems in language processing. HMMs provide a powerful and flexible formalism for modeling sequences of words. They allow us to estimate the posterior probability that a document would be relevant to a user, given the user's query or to compute the probability that a document discusses a particular set of topics. They allow us to determine automatically which words are related to which topics, even though each document is annotated with multiple topics. They even allow us to decompose an unannotated corpus of documents into its component set of topic basis functions. While some of the HMMs used are extremely simple, they afford a paradigm for model parameter estimation and offer the possibility of using more powerful models in the future. In information extraction, we can use HMMs to estimate the probability that a sequence of words in a particular context is a name of a particular type or that two entities in a text are related in a particular way. We describe simple HMMs used for these and other language processing tasks.

8.1 Introduction

HMMs have been used for the last decade as the preferred method for speech recognition. This is because they provide a simple and flexible mechanism for modeling sequences of variable length. The observation sequence in speech is a high-

dimensional vector that is hard to visualize. Although we know the conditional independence assumptions that define the HMM are not really true for speech, the overwhelming advantages of using a rigorous probabilistic formalism still result in high accuracy. Over the years, researchers have found various ways of modeling the dependence to some degree.

In contrast to speech recognition, problems in text processing, at first glance, seem deceptively simple. We can see the words, and we quickly develop many theories as to how words convey meaning and fit together to construct more complex meanings. It is often possible to build primitive systems with a small number of ad hoc rules. Therefore, there is often greater resistance to using probabilistic methods for text problems where rule-based methods seem more intuitive.

In this chapter, we show that the same advantages that hold for speech also apply to other language problems. We discuss two different classes of language processing problems. In the first class, we perform an operation on a whole document. A document is any sizable unit of text, such as a story, a message, etc. This includes, for example, information retrieval (retrieving documents in response to some query) and topic classification (assigning one or more categories from a fixed set to a document). The second class of problems is commonly called Information Extraction. Here we attempt to understand the meaning of some of the text in some way. This includes, for example, extracting parts of speech, detecting and categorizing names, or detecting relations among entities described in the text. The former problem is traditionally performed using one of several similarity measures that have been developed over the years. The latter is traditionally performed using a set of hand constructed rules.

We contrast the use of HMMs with two other common approaches used for language processing: rules, and ad hoc similarity measures.

8.2 Use of Probabilities

This short section discusses the advantages and disadvantages of using probabilities as the basic scoring mechanism for solving language processing problems. Bayes decision theory tells us that if we have to make a decision or choice among several possibilities, and if we want to minimize the probability of error we must compute the probability that each possibility is true given the data and then choose the one with the highest (posterior) probability. So why is there a question? Because the devil is in the details. It is rare that we can actually compute the correct probability of each alternative. Instead, we must make a model with parameters that are possible to estimate using the available data and that are also practical to compute with reasonable resources. In choosing a model that is practical, we often make several simplifying assumptions that ultimately prevent us from computing the true posterior probability. If these assumptions are sufficiently bad, then the resulting errors in the probabilities we obtain can more than offset the benefit for using probabilities in the

first place.
Probabilistic methods almost always come with a model for the problem. If we have a good model for the problem, then we also may get other advantages, such as methods that learn from examples, rather than methods that need to be programmed explicitly. But a short program (or set of rules) can sometimes do better than a bad probabilistic model. So the first answer to the question of whether we should use probabilistic models is, "It depends on how careful we are in using probabilities." We have found that if one is careful, it is usually possible to obtain high performance (comparable to or exceeding the state-of-the-art) using these methods. Furthermore, with the solid mathematical basis for what is being done, we can often analyze what is wrong with our models (usually the assumptions) and improve on them if desired.

8.2.1 Hidden Markov Models

If we accept that we should use probabilistic models, why do we use hidden Markov models (HMMs) for so many different problems? It may seem to some like the only tool we have is the proverbial hammer, so everything looks to us like a nail. While this might be true to some extent, HMMs are appropriate for modeling noisy sequences. All language problems (speech, text, writing) consist of sequences. The sequences are obviously noisy, or else it would be simple to write programs to solve all our language problems.
The most serious problems that we face with HMMs is the conditional independence assumption. The transition from one state to the next depends only on the state, and not on how we get there, how long we have been there, or what symbols were previously emitted while we were in that state. Similarly, the symbol to be emitted from a state depends only on being at that state, and nothing about the history, including previous states or previously emitted symbols. This independence assumption is patently false for most language problems. In speech we know that there is a high correlation among successive spectra. In text, we know that each word depends to a very large extent on the preceding words, both immediately preceding and further back.
So why do these models work so well? First, most of the dependence that we ignore can be thought of as redundant positive correlation. That is, if we treat each observation as new and independent, the probability that we compute for it may be lower than it should be, had we taken into account the previous observations. But this error may be thought of as relatively uniform across most choices. And the benefits for using HMMs are large. They afford us with a well-established set of methods and mathematics for manipulating problems in the way we need to.
Probabilistic models in general, and HMMs in particular, have several advantages. First, as stated above, they provide us with a well-defined mathematical approach to solving pattern recognition problems in language. Second, they often provide us with simple methods for developing models on new domains or languages, requiring only a set of data with annotated answers. Finally, many language problems that do not use probabilistic models use a sum of scores. For example, the conventional metric for comparing documents for Information Retrieval sums up a score for each

word in the document that matches a query word. In order for this sum to be an appropriate measure of relevance, these scores must be log probabilities.

In the remaining sections we discuss several problems in text processing: name spotting, topic classification, information retrieval, event tracking and unsupervised topic discovery. The first problem is different from the others in that it detects and categorizes particular intervals of text as names. The other applications make decisions at the level of a whole document.

In each of these applications, we would like to operate on the output of an automatic speech recognition system. This can present special problems, since the speech recognition output has errors. In addition, the output of a speech recognizer typically does not have any sentence boundaries, case information, or punctuation. Nevertheless, we have found that these techniques work quite well on the output of a speech recognizer. For name spotting, the total error rate is typically the sum of the errors of the speech recognition system and the name spotting error on normal text. For the other problems that operate at the level of a whole document, there is typical no measurable degradation due to speech recognition, despite the fact that 20% of the words may be wrong and the punctuation, case, and sentence boundaries are lost.

8.3 Name Spotting

The objective of name spotting is to extract important terms from the speech and collect them in a database. For example, in news, it is useful to locate names of persons, places, and organizations. Most of the previous work in this area has considered only text sources of written language and has concentrated on the design of rule-driven algorithms to locate the names. Extraction from automatic transcriptions of spoken language is more difficult than written text due to the absence of capitalization, punctuation, and sentence boundaries, as well as the presence of recognition errors. These have significant degrading effects on the performance of rule-driven systems. To overcome these problems, we have developed an HMM-based name extraction system called IdentiFinder [1]. The technique requires only that we provide training text with the type and location of the named entities marked. The system has the additional advantage that it is easily ported to other languages, requiring only a set of annotated training data from a new language.

The name spotting problem can be redefined as having to identify the type of all the words in a document. We must find all examples of names of people, places, and organizations. The remaining text must be correctly classified as not belonging to any of these types. The model that we use reflects this task. The model consists of one state for each of the three named entities plus one state (general language) for all other words in the text, with transitions from each state to every other state. Associated with each of the states is a bigram statistical model on all words in the vocabulary. A different bigram model is estimated for each of the states. By thinking

of this as a generative model that generates all the words in the text, most of the time we are in the GL state emitting general language words. We then transition to one of the named-entity states if we want to generate a name; we stay inside the state generating the words for that name. Then, we either transition to another named-entity state or, more likely, back to the GL state. The decision to emit each word or to transition to another state depends on the previous word and the previous state. In this way the model uses context to help detect and classify names. For example, the word "Mr." in the GL state is likely to be followed by a transition to the PERSON state. After the person's name is generated, a transition to the GL state is likely and general words like "said" or "departed" may follow. These context-dependent effects are included in our model.

The parameters of the model are estimated automatically from annotated training data, where the three sets of named entities are marked in the text. Then, given a test sample, the model is used to estimate the probability of each word belonging to one of the three named entities or to none. We then use the Viterbi algorithm [2] to find the most likely sequence of states to account for the text. The result is the answer for the sequence of named entities.

Since our system has been trained on only one million words of annotated data from broadcast news, many of the words in an independent test set will be unknown to the name spotting system, even though they might be known to the speech recognizer. (Words that are not known to the speech recognizer will be recognized incorrectly as one of the existing words and will, of course, cause performance degradation, as we shall see below.) It is important to deal with the unknown word problem since some of those words will be among the desired named entities, and we would like the system to spot them even though they were not seen before by the training component. During training, we divide the training data in half. In each half we replace every string that does not appear in the other half with the string 'UNKNOWN'. We then are able to estimate all the probabilities involving unknown words. The probabilities for known words are estimated from all of the data. During the testing phase, we replace any string that is unknown to the name spotting system by the label 'UNKNOWN,' and we are then able to find the best matching sequence of states. We have found that, by making proper use of context, more than half of the names that were not known to the name spotting system are labeled correctly by the system.

One advantage of our approach to information extraction is the ease with which we can learn the statistics for different styles of text. For example, let us say we want the system to work on text without case information (i.e., the text is displayed as either all lower case or all upper case). It is a simple matter to remove the case information from our annotated text and then reestimate the models. If we want to use IdentiFinder on the output of a speech recognizer, we expect that the text will not only be caseless, but will also have no punctuation. In addition, there will be no abbreviations, and numeric values will be spelled out (e.g., TWENTY FOUR rather than 24). Again, we can easily simulate this effect on our annotated text in order to learn a model of text output from a speech recognizer. Of course, given annotated data from a new language, it is a simple matter to train the same system to recognize named entities in that language.

We have performed several experiments to measure the performance of IdentiFinder in finding names. In addition, we have measured the degradation when case and punctuation information is lost, or when faced with errors from automatic speech recognition. In measuring the accuracy of the system, both the type of named entity and the span of the corresponding words in the text are taken into consideration. We measure the slot error rate where the type and span of a name is each counted as a separate slot by dividing the total number of errors in named entities (substitutions, deletions, and insertions) by the total number of true named entities in the reference answers [3].

In a test from the DARPA Broadcast News corpus [4], where the number of types of named entities was seven (rather than the three used here), IdentiFinder obtained a slot error rate of 11.4% for text with mixed case and punctuation. When all case and punctuation were removed, the slot error rate increased to only 16.5%.

In recent DARPA evaluations on name spotting with speech input, again with seven classes of names, the slot error rate for the output of the Byblos speech recognizer was 26.7% with a speech recognition word error rate of 14.7% [5]. When all recognition errors were corrected, without adding any case or punctuation information, the slot error rate decreased to 14.1%. In general, we have found that the named-entity slot error rate increases linearly with the word error rate in approximately a one-to-one fashion.

8.4 Topic Classification

Much work has been done in topic classification where the models for the different topics are estimated independently, even if multiple topics are assigned to each document. One notable exception is the work of Yang and Chute [6] who, as part of their model, take into consideration the fact that multiple simultaneous topics are usually associated with each document. Our approach to topic classification is similar in spirit to that of Yang and Chute, except that we use a Bayesian framework [7] instead of a distance-based approach. Our topic classification component, called OnTopic^{TM}, is a probabilistic HMM whose parameters are estimated from training samples of documents with given topic labels, where the topic labels number in the thousands. The model allows each word in the document to contribute different amounts to each of the topics assigned to the document. The output from OnTopic is a rank-ordered list of all possible topics and corresponding scores for any given document.

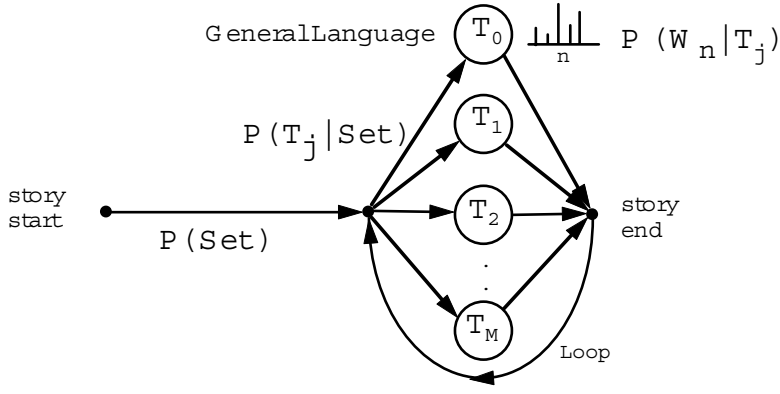

FIGURE 8.1
A hidden Markov model for topics. Each state can emit words for one topic. State T0 emits words corresponding to general language.

8.4.1 The Model

We choose the set of topics Set that corresponds to a given document D such that the posterior probability $P(Set \mid D)$ is maximized:

$$P(Set \mid D) = P(Set)\frac{P(D \mid Set)}{P(D)}. \tag{8.1}$$

For the purpose of ranking the sets of topics, $P(D)$ can be ignored. The prior probability is really the joint probability of a document having all the labels in the set, which can be approximated using topic co-occurrence probabilities:

$$P(Set) \approx \left[\prod_{\{k \in Set\}} \prod_{\{m \in Set,\, m>k\}} P(T_k, T_m) \right]^{\frac{1}{\binom{N}{2}}} \tag{8.2}$$

where N is the number of topics in *Set* and the exponent serves to place on similar footing topic sets of different sizes. $P(T_k, T_m)$ is estimated by taking the product of the maximum likelihood estimates of $P(T_k \mid T_m)$ and $P(T_m)$. The former is estimated as the fraction of those documents with T_m as a topic which also have T_k as a topic, and the latter is estimated as the fraction of documents with T_m as a topic.

What remains to be computed is $P(D \mid Set)$, the conditional probability of the words in the document, given that the document is labeled with all the topics in Set.

We model this probability with an HMM consisting of a state for each of the topics in the set, plus one additional topic state, general language (GL), as shown in Fig. 8.4.1. The model "generates" the words in the document one by one, first choosing a topic distribution from which to draw the next word, according to $P(T_j \mid Set)$, then choosing a word according to $P(W_t \mid T_j)$, then choosing another topic distribution to draw from, etc. The formula for $P(D \mid Set)$ is, therefore,

$$P(D \mid Set) \approx \prod_t \sum_{j \in Set} P(T_j \mid Set) P(W_t \mid T_j) \tag{8.3}$$

where t varies over the set of words in the document. The elements of the above equation are estimated from training data as described below.

8.4.2 Estimating HMM Parameters

We use a biased form of the Expectation-Maximization (EM) algorithm [8] to find good estimates for the transition probabilities and the emission probabilities $P(T_j \mid Set)$ in the HMM in Fig. 8.4.1. The transition probabilities are defined by:

$$P_{k+1}(T_j \mid Set) = \frac{E(\text{\# times any word is emitted in state } T_j \mid \text{model } k)}{E(\text{\# times any word is emitted in any state} \mid \text{model } k)} \tag{8.4}$$

which can be estimated as

$$P_{k+1}(T_j \mid Set) = bias(T_j) \frac{\sum_D \sum_{W \in D} q_{k,j}(W, T_j, D)}{\sum_i \sum_D \sum_{W \in D} q_{k,i}(W, T_i, D)} \tag{8.5}$$

where

$$bias(T_j) = \frac{\sum_{D \text{ with } T_j} l(D)}{\sum_D l(D)} \tag{8.6}$$

is the bias term, $l(D)$ is the number of words in the document D, and

$$q_{k,j}(W, T_j, D) = c(W \mid D) I(D \text{ has } T_j) \frac{P_k(T_j \mid Set) P_k(W \mid T_j)}{\sum_{i \in Set} P_k(T_i \mid Set) P_k(W \mid T_i)}. \tag{8.7}$$

$q_{k,j}(W, T_j, D)$ is the fraction of the counts for W in D that are accounted for by T_j, given the current set of parameters in the generative model; $c(W \mid D)$ is the number of times that word W appears in the document; and $I(x)$ is an indicator function returning 1 if its predicate is true and 0 otherwise. The bias term is needed to bias the observations towards the GL state; otherwise, the EM algorithm would result in a zero transition probability to the GL state [6]. The effect of the bias is that the transition and emission probabilities for topic will be set such that this topic accounts for a fraction of the words in the corpus roughly equal to $bias(T_j)$. The emission probabilities are then estimated from:

$$P_{k+1}(W \mid T_j) = \frac{\sum_D q_{k,j}(W, T_j, D)}{\sum_D \sum_{W \in D} q_{k,j}(W, T_j, D)}. \tag{8.8}$$

8.4.3 Classification

To perform classification for a given document, we need to find the set of topics that maximizes (8.1). But the total number of all possible sets is ($\sum_{k=1}^{M} k!$), which is a very large number if the number of possible topics M is in the thousands. Since scoring such a large number of possibilities is prohibitive computationally, we employ a two-pass approach. In the first pass, we select a small set of topics that are likely to be in the best set. In the second pass, we score all sets of these candidates using (8.1). We select candidate topics in the first pass by scoring each topic independently, as if it were a complete set on its own, using a slight modification of (8.1):

$$\log P(T_j \mid D) \approx \alpha \log P(T_j) + \sum_t \phi \left(\log \left[P(T_j \mid Set)^{\beta} \frac{P(W_t \mid T_j)}{P(W_t)} \right] \right) \quad (8.9)$$

where $\phi(x)$ is 0 if $x < 0$ and x otherwise, and serves to filter out the effect of words in documents that constitute negative evidence for a topic. The parameter α has been introduced to balance the prior against the generative model and is optimized from training data. The parameter β is there to flatten (if less than one) or sharpen (if greater than one) the transition probability distribution, in order to compensate for the independence assumption over words in the document.

8.4.4 Experiments

We applied the two-pass procedure of the OnTopic classifier described above to a corpus of broadcast news stories, transcribed and annotated by Primary Source Media. For each story, the annotators gave a number of topic labels that they thought representing the topics in the story. The number of topics for each story was anywhere between 1 and 13, with an average of 4.5 topics per story. The corpus was divided into one year, or 42,502 stories, for training, and one month, or 989 stories, for test. The training set contained a total of 4627 unique topic labels.

Measuring the performance of our system against what the human annotators wrote down as the topic labels is not straightforward, because our system gives an ordered list of all topics, each with a score, while the annotators have a small, unordered list of topics for each story. We measure the performance as a function of the number N of top-ranking topics provided by the system. For each value of N, we compare the top-N topics produced by the system against the set of topics generated by the annotators. The accuracy was 76% for the first choice and decreased to about 50% for the fifth choice.

We have indications that the criteria we have adopted for measuring the performance of our system may be less forgiving than necessary. Topic annotation is not an easy task for people when the number of topics is large; people tend to under-generate labels for documents because it is difficult to remember so many topics. Upon informal examination of stories for which the top-scoring topic was not included in the list given by the annotators, we found that well over 90% of the time, the topic given by the computer was quite reasonable for the story. In these cases, the human

annotators had simply not been exhaustive in their enumeration of the possible topics for the story.

8.5 Information Retrieval

Information retrieval is the task of finding documents that are relevant to a query, where that query might contain a small number of words typed by a person, or might even consist of several documents indicated as interesting. To perform this task, we developed an information retrieval (IR) system, called Golden Retriever [9]. Golden Retriever is a novel probabilistic HMM-based IR system that computes the probability that a document is relevant, given a query, and ranks all documents in the collection based on this measure. Our approach to IR mirrors our topic classification work; we allow a corpus of examples to drive our selection of models and our estimation procedures. The corpus consists of a set of documents, a set of natural language queries (tens of words), and a number of relevance judgments that state whether each document is relevant to the query or not. Human annotators make the relevance judgments on some significant sampling of the corpus of documents for each query. We build a statistical model capable of ranking training documents effectively by their labeled relevance to given training queries.

8.5.1 A Bayesian Model for IR

Given a query, it seems sensible to rank the documents in a corpus by their probability of being relevant [10]. In other words, we want to use as our document ranking function the posterior probability $P(D \in R \mid Q)$, the probability that the document D is relevant, given query Q. We again use Bayes' rule to decompose the posterior probability:

$$P(D \in R \mid Q) = \frac{P(D \in R)P(Q \mid D \in R)}{P(Q)} \tag{8.10}$$

$P(D \in R)$ is the prior probability of a document being relevant to any query. While this might not sound meaningful, it can be a very powerful source of information. For example, documents from particular sources are more likely to be useful. Newer documents may be more relevant than old ones, long documents more relevant than short ones, and so on. In addition, there are likely to be particular documents that everyone wants. These preferences can be universal, or a function of the group the user is in, or even particular to the individual user. $P(Q)$ is simply the prior probability of the query being posed in the first place. As this quantity does not alter the document ranking, we can safely ignore it. What is left is the conditional probability of the query being posed, under the hypothesis that the document is relevant, $P(Q \mid D \in R)$. We model this remaining quantity with a discrete HMM that is dependent on the document. This will be a generative model where we think of the

document HMM as generating the query. The parameters of the HMM should be estimated in such a way as to make it more likely that a document will generate a query to which it is relevant than a query to which it is not relevant.
A simple formulation of the requisite HMM has just two states labeled D and GL, with state D representing the option of generating query words by drawing words directly from the document and state GL representing choosing words from general language, i.e., without regard to the document. Most queries contain words that are present in relevant documents, but all queries contain many general words that are not really part of the specification of relevant documents.

8.5.2 Training the IR HMM

The parameters of the HMM are the transition probability α (the probability that the query word will be chosen from the GL state), and the emission probabilities for each of the words in each state. In principle, we would like to estimate these parameters from examples using the EM algorithm. In practice, however, we find that we do not have enough training examples to find good estimates for the emission probabilities. So we set the emission probabilities for the D and GL states to be the unigram distributions of the words in the document and the whole corpus, respectively. Further, we set the transition probabilities to be the same α for all documents, and we estimate α using the EM algorithm. We found that the value of α depends on the type of query. If the query is a short phrase typed by a human, then a typical value for α is 0.3, meaning that 70% of the words in the query would be expected to be found in any relevant document. But if the query were a long description, then the typical value of α was 0.7, meaning that only 30% of the words would be expected to appear in relevant documents. Finally, if the query were an entire document, even higher values of α would be appropriate.

8.5.3 Performance

We have tested this simple two-state HMM on the TREC-7 (Text Retrieval Conference) corpus which consists of 528,155 documents [9]. We preprocess the corpus lightly in order to split documents up into words and to allow morphologically similar words to match. The stream of characters in each document gets tokenized into words. Then we conflate terms by applying Porter's stemming algorithm [11]. Next, we discard anything found in a list of 400 "stop" words. Finally, numeric and non-word items are reduced to single tokens ('NUMBER', 'DOLLAR', etc.).
The test comprised 50 queries with an average of 57.6 words per query. Each query was preprocessed in the same manner described above for the documents. Then, for each query we compute (8.10) for each of the documents. The result is that the top-scoring document for each query was found to be relevant 78% of the time.
The simple model described above has been extended by adding more states with different query term-generating mechanisms (e.g., synonyms, bigrams, topics, unsupervised relevance feedback), and by the inclusion of document priors, resulting in higher performance [12].

8.6 Event Tracking

Another application of topic classification is called event tracking. In this case, we are given a small number of documents that describe an event and we are asked to find all of the other documents that describe the same event. The Topic Detection and Tracking Project sponsored by the government posed just such a problem. In this problem, we are given four documents that are purported to be about a particular event. Our task is to find the remaining documents about the same event. But there are some complications in the task. First, we must make a binary decision for each document as soon as we look at it. We cannot look at the whole corpus in order to rank the documents. But we are also given a part of the corpus and told that most of these documents do not discuss the same event. Thus, we must devise a mechanism for determining a threshold on the score in order to know whether or not to accept the document. A second part of the problem is that the nature of the documents is likely to change over time as the discussion of the original event evolves. For example, the initial stories might be about the commission of some crime. The later stories might be about the trial of the person(s) accused of that crime.

One can think of this problem in two ways. First, we can think of the example documents as defining a topic. We can estimate a topic model (i.e., word distribution) from these example documents. Then, our task is to find those documents in the corpus that have a high probability of discussing that same topic. Alternatively, we can think of the example documents as a (very long) query. And we would then be looking for documents that are relevant to us, given that the query was as given. Note that these two formulations result in entirely different models and processes. In the first case, we compute a topic model from the example documents, and then, using Bayes' rule, we compute the probability of the words in each document, given that the document is about this topic. In the second case, we compute a distribution from each document and then compute the probability of the given 'query' given this document is relevant.

The topic classification mechanism has some advantages. For example, as we go through stories, we find several stories that have a very high likelihood of being about the same event. We can, if sufficiently confident, add these documents into the model as if they were training examples for the event. Then, we retrain our model for the topic and proceed from there. The information retrieval model cannot do this as easily. That is because we are imagining that all of the training documents were 'generated' by each relevant document. But at the point where the number of training documents becomes very much larger than each later document, we cannot really expect the single document to contain most of the words in the training documents.

The problem of deciding whether a document is relevant to the four example documents (as opposed to just ranking the documents in the corpus) is a difficult one. This is because the scores for event model and document will vary as a function of unimportant factors. For example, for the topic classification model, the number word scores for a test document is a function of the length of the document. While

we can compute the average score per word, the score still depends on the particular words that appear in the document. On the other hand, the Information Retrieval score will be relatively unbiased for the different documents, but the score will vary as a function of the length of the query (the four example documents, in this case), and the particular words used in those documents.

To solve this problem, we added another level of classifier. Given a model for a topic, we can compute the score for a large number of documents in our corpus. We remove those documents that we either know to be on the same event or receive a very high score (they may be about the same event). Then, we make a distribution of those scores. We can then set a threshold based on that distribution. For example, we typically set a threshold at six standard deviations above the mean score value. (The distribution of scores is not really Gaussian. We just use this mechanism as a convenient one.) The result is that some fixed fraction of off-topic stories would be incorrectly classified as being on the event. That is, we fix the probability of false acceptance of a story that is not about the desired event. This results in extremely consistent behavior of the classifier across different events and different documents. The resulting probability of missing a document that is on the event is whatever comes from the nature of the problem. But the number of false alarms is completely predictable.

8.7 Unsupervised Topic Detection

In order to use topic classification, we presume that we have a large set of documents annotated with topics. But, for a large number of topics we need an even larger corpus. The cost of annotating a large corpus with topics can be quite large. Furthermore, the topics continue to change over time, requiring constant annotation. Finally, we must do this for every domain and every language.

However, it is possible to annotate a corpus automatically with topics similar to those that would be created by human annotators. To do so, we use the nature of topics, and we also use the tools that we created for the supervised topic classification problem.

First, we can abstract the definition of topics to be a collection of sets of words. The words within each set are statistically related. We assume documents are written by drawing from these sets of words, somewhat at random. Second, we need to have understandable names for each of these topics, otherwise it is not clear what a person would use them for.

We have devised a method for deriving a large set of named topics from a corpus of documents. The method works in any language or domain. We have evaluated the topics and found that they are comparable in their behavior to the topics assigned by human annotators and, for some applications, even result in superior performance to human-created topic sets.

Before we start, we derive various phrases from the corpus, since phrases often carry

more obvious meaning than single words. We use two techniques to find phrases in the corpus. The first one is simply to find strings of words that are "sticky". That is, the words tend to appear together much more often than otherwise expected. There are several techniques that could be used for this. For example, one can use the mutual information between words. We have used a minimum description length technique that finds those strings, which, if adopted, most reduce the number of bits needed to encode the entire corpus. The second technique for finding phrases uses the named-entity extraction technique described in the previous section. This assumes that some text in the language has been annotated with examples of names of different types. (Here, names can be generalized to include any types of interesting phrases.) For example, for news, we found those phrases that were names of people, locations, or organizations. These phrases are used simply to define additional terms. The original words are kept, but additional strings are added for the phrases. Next, the algorithm chooses candidate topic names. These are chosen simply as words or phrases that seem to have high information content. Specifically, for each term (word or phrase) in the document, we compute its inverse document frequency as

$$Idf(w) = \log(N/df(w))$$

where N is the total number of documents in the corpus, and df(w) is the number of documents in the corpus that contain the word w at least once. Then, within each document, we compute the number of times each word occurs, multiplied by the IDF of that term. The result is an ad hoc measure of how likely that term is to be an important concept for the document.

We choose the highest five such terms for each document, subject to the condition that each of these terms occurs some minimum number of times in the entire corpus. (We have used 4 as a threshold.) This then means that this term is a plausible name of a topic, and we also have a set of documents that seem to contain the term. These are taken to be to be initial topics and topic assignments. If we change these thresholds then the number of topics produced by the system will vary.

But our goal is to find full topics, where topics are probability distributions of large sets of words, and to assign these topics to all the documents that discuss them - not just those that contain the name of the topic. So we use the same OnTopic training algorithm described in above. That is, we take the initial topic assignments to be correct. We estimate new models for the topics. This has the effect of adding the other content words from the documents to the topic models defined. Then, we use these topic models to classify all of the documents. The result is that the topics are now assigned to many more documents, many of which do not contain the names of the topics.

We have used this algorithm on the same corpus from Primary Source Media described above. In this case, we created about 18,000 topics automatically. The topics created were judged by presenting the chosen topics for several documents to subjects, who decided whether each one was appropriate. We found that on over 90% of the documents, the first choice document seemed relevant to the document. Furthermore, on the fifth choice topic, the relevance was still over 80%. This result was

quite high in that it was comparable to, or even higher than that of the topic models trained from human annotation.

The great advantage of unsupervised topic discovery is that it can easily be applied to any new domain or language. It can also find very large numbers of topics. In fact, we have applied it to Arabic news stories and found that the performance was comparable to that on English stories. This made it possible to perform topic classification on the Arabic news stories, since it would have been far too expensive for us to annotate tens of thousands of news stories with topics manually.

8.8 Summary

We have described several different text processing techniques that can be performed using HMMs. In most cases, the resulting algorithms were as good as or better than the existing techniques for performing that function. Furthermore, the techniques could be used trivially in any language or domain. Finally, we have seen that the techniques can often be extended to result in higher accuracy or to perform related functions relatively easily, because of their underlying simplicity.

References

[1] D.M. Bikel, R. Schwartz, and R.M. Weischedel, "An Algorithm that Learns What's in a Name," Machine Learning, Kluwer Academic Publishers, Vol. 34, pp. 211-231, 1999.

[2] G.D. Forney, "The Viterbi Algorithm," Proc. IEEE, Vol. 61, pp. 268-278, 1973.

[3] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance Measures for Information Extraction," DARPA Workshop on Broadcast News Understanding, Herndon, VA, Morgan Kaufmann Publishers, pp. 249-252, March 1999.

[4] The data used in the various experiments reported in this paper is available from the Linguistic Data Consortium (LDC), University of Pennsylvania, http://www.ldc.upenn.edu/.

[5] D. Miller, R. Schwartz, R. Weischedel, and R. Stone, "Named Entity Extraction from Broadcast News," DARPA Workshop on Broadcast News Understanding, Herndon, VA, Morgan Kaufmann Publishers, pp. 37-40, March 1999.

[6] Y. Yang and C.G. Chute, "An Example-Based Mapping Method for Text Cat-

egorization and Retrieval," ACM Trans. Information Systems, Vol. 12, No. 3, pp. 252-277, July 1994.

[7] R. Schwartz, T. Imai, F. Kubala, L. Nguyen, and J. Makhoul, "A Maximum Likelihood Model for Topic Classification of Broadcast News," Eurospeech '97, Rhodes, Greece, pp. 1455-1458, Sept. 1997.

[8] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," J. Royal Statistical Society B, Vol. 39, No. 1, pp. 1-22, 1977.

[9] D. Miller, T. Leek, and R. Schwartz, "BBN at TREC7: Using Hidden Markov Models for Information Retrieval," The Seventh Text Retrieval Conference (TREC-7), NIST Special Publication 500-242, pp. 133-142, July 1999.

[10] M.E. Maron and K.L. Kuhns, "On relevance, probabilistic indexing, and information retrieval," J. Assoc. Computing Machinery, Vol. 7, pp. 216-244, 1960.

[11] M.F. Porter, "An Algorithm for Suffix Stripping," Program, Vol. 14, No. 3, pp. 130-137, 1980.

[12] D.R.H. Miller, T. Leek, and R.M. Schwartz, "A Hidden Markov Model Information Retrieval System," 22nd Int. ACM SIGIR Conf. Research and Development in Information Retrieval, Berkeley, CA, pp. 214-221, Aug. 1999.

9

Statistical Language Models With Embedded Latent Semantic Knowledge

Jerome R. Bellegarda
Apple Computer, Inc.

CONTENTS

9.1 Introduction

The Bayesian approach pervasive in today's speech recognition systems entails the construction of a prior model of the language, as pertains to the domain of interest. The role of this prior, in essence, is to quantify which word sequences are acceptable in a given language for a given task, and which are not. It must therefore encapsulate as much as possible of the syntactic, semantic, and pragmatic characteristics of the domain [35, 50]. In the past two decades, it has become increasingly common to do so through statistical n-gram language modeling (LM), where each word is predicted conditioned on the current context, on a left to right basis [16, 53]. Although widespread, this solution is not without drawbacks: prominent among the challenges faced by n-gram modeling is the inherent locality of its scope, due to the limited amount of context available for predicting each word.

9.1.1 Scope Locality

Central to this issue is the choice of n, which has implications in terms of predictive power and parameter reliability. Large values of n are desirable for the former, but low values of n are necessary for the latter (see, for example, [39, 46, 47]). This in

turn imposes an artificially local horizon to the model, impeding its ability to capture large-span relationships in the language.
To illustrate, consider, in each of the two equivalent phrases:

stocks fell sharply as a result of the announcement (9.1)
stocks, as a result of the announcement, sharply fell (9.2)

the problem of predicting the word "*fell*" from the word "*stocks*." In (9.1), this can be done with the help of a bigram LM ($n = 2$), which is straightforward with the kind of resources currently available [51]. In (9.2), however, the value $n = 9$ would be necessary, a rather unrealistic proposition at the present time. Largely because of this inability to reliably capture large-span behavior, n-gram performance has essentially reached a plateau [53].
This observation has sparked interest in a variety of countermeasures, involving for instance *information aggregation* or *span extension* [5]. Information aggregation increases the reliability of a word prediction by taking advantage of exemplars of other words that behave "like" this word in the particular context considered. The trade-off, typically, is higher robustness at the expense of a loss in resolution. This chapter is more closely aligned with span extension, which extends and/or complements the n-gram paradigm with information extracted from large-span units (i.e., comprising a large number of words). The trade-off here is in the choice of units considered for the analysis of long distance dependencies. These units tend to be either syntactic or semantic in nature.

9.1.2 Syntactically–Driven Span Extension

Assuming a suitable parser is available for the domain considered, syntactic information can be used to incorporate large-span constraints into the recognition. How these constraints are incorporated varies from estimating n-gram probabilities from grammar-generated data [62] to computing a linear interpolation of the two models [37]. Most recently, syntactic information has been used specifically to determine equivalence classes on the n-gram history, resulting in so-called dependency [14, 49] or structured [15, 36, 58] LMs. In that framework, each unit is the headword of the phrase spanned by the associated parse sub-tree. The standard n-gram LM is then modified to operate given the last $(n-1)$ *headwords* as opposed to the last $(n-1)$ *words*. As a result, the structure of the model is no longer pre-determined: which words serve as predictors depends on the dependency graph, which is a hidden variable [53]. In the example above, the top two headwords in the dependency graph would be "*stocks*" and "*fell*" in both cases, thereby solving the problem.
The main caveat in such modeling is the reliance on the parser, and particularly the implicit assumption that the correct parse will in fact be assigned a high probability [61]. The basic framework was recently extended to operate efficiently in a left-to-right manner [15, 36], through careful optimization of both chart parsing [59] and search modules. Also noteworthy is a somewhat complementary line of research [60], which exploits the syntactic structure contained in the sentences prior to the one featuring the word being predicted.

9.1.3 Semantically–Driven Span Extension

High level semantic information can also be used to incorporate large-span constraints into the recognition. Since by nature such information is diffused across the entire text being created, this requires the definition of a *document* as a semantically homogeneous set of sentences. Then each document can be characterized by drawing from a (possibly large) set of topics, usually pre-defined from a hand-labelled hierarchy, which covers the relevant semantic domain [34, 55, 56]. The main uncertainty in this approach is the granularity required in the topic clustering procedure [26]. To illustrate, in (9.1) and (9.2), even perfect knowledge of the general topic (most likely, "*stock market trends*") does not help much.

An alternative solution is to use long distance dependencies between word pairs which show significant correlation in the training corpus. In the above example, suppose that the training data reveals a significant correlation between "*stocks*" and "*fell*." Then the presence of "*stocks*" in the document could automatically trigger "*fell*," causing its probability estimate to change. Because word proximity is now irrelevant, the two phrases would lead to the same result. In this approach, the pair (*stocks*, *fell*) is said to form a word trigger pair [45]. In practice, word pairs with high mutual information are searched for inside a window of fixed duration. Unfortunately, trigger pair selection is a complex issue: different pairs display markedly different behavior, which limits the potential of low frequency word triggers [52]. Still, self-triggers have been shown to be particularly powerful and robust [45], which underscores the desirability of exploiting correlations between the current word and features of the document history.

Recent work has sought to extend the word trigger concept by using a more comprehensive framework to handle the trigger pair selection [2, 3, 4, 7, 19, 29, 31]. This is based on a paradigm originally formulated in the context of information retrieval, called *latent semantic analysis* (LSA) [11, 22, 25, 27, 32, 43, 44, 57]. In this paradigm, co-occurrence analysis still takes place across the span of an entire document, but every combination of words from the vocabulary is viewed as a potential trigger combination. This leads to the systematic integration of long-term semantic dependencies into the analysis, as long as there is a way to identify article boundaries in the available training data. This is the case, for example, with the ARPA North American Business (NAB) News corpus [40]. The LSA paradigm can be used for word and document clustering [7, 29, 31], as well as for language modeling [2, 19]. In all cases, it was found to be suitable to capture some of the global semantic constraints present in the language. In fact, hybrid n-gram+LSA LMs, constructed by embedding LSA into the standard n-gram formulation, were shown to result in a substantial reduction in average word error rate [3, 4].

9.1.4 Organization

The focus of this chapter is on semantically-driven span extension only, and more specifically on how to exploit the LSA paradigm to improve statistical LM. The main objectives are: (i) to review the data-driven extraction of latent semantic infor-

mation, (ii) to assess potential usage in spoken language processing, (iii) to describe integration with conventional n-gram LM, (iv) to examine the behavior of the resulting hybrid models in speech recognition experiments, and (v) to discuss factors which influence performance.

The chapter is organized as follows. In the next two sections, we give an overview of LSA feature extraction and the resulting LSA feature space. Section 9.4 explores the applicability of this framework for general semantic classification. In Section 9.5, we shift the focus to LSA-based statistical LM for large vocabulary recognition. Section 9.6 describes the various smoothing possibilities available to make LSA-based LMs more robust. In Section 9.7, we illustrate some of the benefits associated with hybrid n-gram+LSA modeling on a subset of the Wall Street Journal (WSJ) task. Finally, Section 9.8 discusses the inherent trade-offs associated with the approach, as evidenced by the influence of the data selected to train the LSA component of the model.

9.2 Latent Semantic Analysis

Let $\mathcal{V}$, $|\mathcal{V}| = M$, be some underlying vocabulary and $\mathcal{T}$ a training text corpus, comprising N articles (documents) relevant to some domain of interest (like business news, for example, in the case of the NAB corpus [40]). The LSA paradigm defines a mapping between the discrete sets $\mathcal{V}$, $\mathcal{T}$ and a continuous vector space $\mathcal{S}$, whereby each word w_i in $\mathcal{V}$ is represented by a vector $\bar{u}_i$ in $\mathcal{S}$, and each document d_j in $\mathcal{T}$ is represented by a vector $\bar{v}_j$ in $\mathcal{S}$.

9.2.1 Feature Extraction

The starting point is the construction of a matrix (W) of co-occurrences between words and documents. In marked contrast with n-gram modeling, word order is ignored, which is of course in line with the semantic nature of the approach [44]. This makes it an instance of the so-called "bag-of-words" paradigm, which disregards collocational information in word strings: the context for each word essentially becomes the entire document in which it appears. Thus, the matrix W is accumulated from the available training data by simply keeping track of which word is found in what document.

This tends to involve some appropriate function of the word count in each document [7]. Various implementations have been investigated by the information retrieval community (see, for example, [24]). Evidence points to the desirability of normalizing for document length and word entropy. Thus, a suitable expression for the (i, j) cell of W is:

$$w_{i,j} = (1 - \varepsilon_i) \frac{c_{i,j}}{n_j}, \tag{9.3}$$

where $c_{i,j}$ is the number of times w_i occurs in d_j, n_j is the total number of words present in d_j, and ε_i is the normalized entropy of w_i in the corpus $\mathcal{T}$. The global weighting implied by $1 - \varepsilon_i$ reflects the fact that two words appearing with the same count in d_j do not necessarily convey the same amount of information about the document; this is subordinated to the distribution of the words in the collection $\mathcal{T}$.
If we denote by $t_i = \sum_j c_{i,j}$ the total number of times w_i occurs in $\mathcal{T}$, the expression for ε_i is easily seen to be:

$$\varepsilon_i = -\frac{1}{\log N} \sum_{j=1}^{N} \frac{c_{i,j}}{t_i} \log \frac{c_{i,j}}{t_i} \,. \tag{9.4}$$

By definition, $0 \leq \varepsilon_i \leq 1$, with equality if and only if $c_{i,j} = t_i$ and $c_{i,j} = t_i/N$, respectively. A value of ε_i close to 1 indicates a word distributed across many documents throughout the corpus, while a value of ε_i close to 0 means that the word is present only in a few specific documents. The global weight $1 - \varepsilon_i$ is therefore a measure of the indexing power of the word w_i.

9.2.2 Singular Value Decomposition

The $(M \times N)$ word-document matrix W defines two vector representations for the words and the documents. Each word w_i can be uniquely associated with a row vector of dimension N, and each document d_j can be uniquely associated with a column vector of dimension M. Unfortunately, this is unpractical for three related reasons. First, the dimensions M and N can be extremely large; second, the vectors w_i and d_j are typically very sparse; and third, the two spaces are distinct from one another.
To address these issues, one solution is to perform the (order-R) singular value decomposition (SVD) of W as [30]:

$$W \approx \hat{W} = U S V^T \,, \tag{9.5}$$

where U is the $(M \times R)$ left singular matrix with row vectors u_i $(1 \leq i \leq M)$, S is the $(R \times R)$ diagonal matrix of singular values $s_1 \geq s_2 \geq \cdots \geq s_R > 0$, V is the $(N \times R)$ right singular matrix with row vectors v_j $(1 \leq j \leq N)$, $R \ll \min(M, N)$ is the order of the decomposition, and ^{T} denotes matrix transposition. As is well known, $\hat{W}$ is the best rank-R approximation to the word-document matrix W, for any unitarily invariant norm (cf., e.g., [20]). Furthermore, both left and right singular matrices U and V are column-orthonormal, i.e., $U^T U = V^T V = I_R$ (the identity matrix of order R). Thus, the column vectors of U and V each define an orthornormal basis for the space of dimension R spanned by the u_i's and v_j's.
Upon projecting the row vectors of W (i.e., words) onto the orthornormal basis formed by the column vectors of V, the row vector $u_i S$ characterizes the position of word w_i in the underlying R-dimensional space, for $1 \leq i \leq M$. Similarly, upon projecting the column vectors of W (i.e., documents) onto the orthonormal basis formed by the column vectors of U, the row vector $v_j S$ characterizes the position of document d_j in the same space, for $1 \leq j \leq N$. We refer to each of the

M scaled vectors $\bar{u}_i = u_i S$ as a *word vector*, uniquely associated with word w_i in the vocabulary, and each of the N scaled vectors $\bar{v}_j = v_j S$ as a *document vector*, uniquely associated with document d_j in the corpus. Thus, (9.5) defines a transformation between high-dimensional discrete entities ($\mathcal{V}$ and $\mathcal{T}$) and a low-dimensional continuous vector space $\mathcal{S}$, the R-dimensional (LSA) space spanned by the u_i's and v_j's.

The dimension R is bounded from above by the (unknown) rank of the matrix W, and from below by the amount of distortion tolerable in the decomposition. It is desirable to select R so that $\hat{W}$ captures the major structural associations in W and ignores higher order effects. In addition, classical methods for determining the SVD of dense matrices (see, for example, [12]) are not optimal for large sparse matrices such as W. Instead, it is more appropriate to solve a sparse symmetric eigenvalue problem, which can then be used to indirectly compute the sparse singular value decomposition. Several suitable iterative algorithms have been proposed by Berry, based on either the subspace iteration or the Lanczos recursion method [10]. Convergence is typically achieved after 100 or so iterations.

9.2.3 General Behavior

By construction, the "closeness" of vectors in the LSA space $\mathcal{S}$ is determined by the overall pattern of the language used in $\mathcal{T}$, as opposed to specific constructs. Hence, two words whose representations are "close" (in some suitable metric) tend to appear in the same kind of documents, whether or not they actually occur within identical word contexts in those documents. Conversely, two documents whose representations are "close" tend to convey the same semantic meaning, whether or not they contain the same word constructs. We can therefore expect words and documents that are semantically linked to also be "close" in the LSA space $\mathcal{S}$.

Of course, the optimality of this framework can be debated, since the underlying L_2 norm may not be the best choice when it comes to linguistic phenomena. For example, the Kullback-Leibler divergence provides a more elegant (probabilistic) interpretation of (9.5) [32], albeit at the expense of requiring a conditional independence assumption on the words and the documents [33]. This caveat notwithstanding, the correspondence between closeness in LSA space and semantic relatedness is well documented. In applications such as information retrieval, filtering, induction, and visualization, the LSA framework has repeatedly proven remarkably effective in capturing semantic information [11, 22, 25, 27, 33, 43, 44, 57].

Such behavior was recently illustrated in [48], in the context of an (artificial) information retrieval task with 20 distinct topics and a vocabulary of 2000 words. A probabilistic corpus model generated 1000 documents, each 50 to 100 words long. The probability distribution for each topic was such that 0.95 of its probability density was equally distributed among topic words, and the remaining 0.05 was equally distributed among all the 2000 words in the vocabulary. A suitable distance was then measured between all pairs of documents, both in the original space and in the LSA space obtained as above, with $R = 20$. This leads to the expected distance distributions depicted in Figure 9.1, where a pair of documents is considered "intra-topic" if

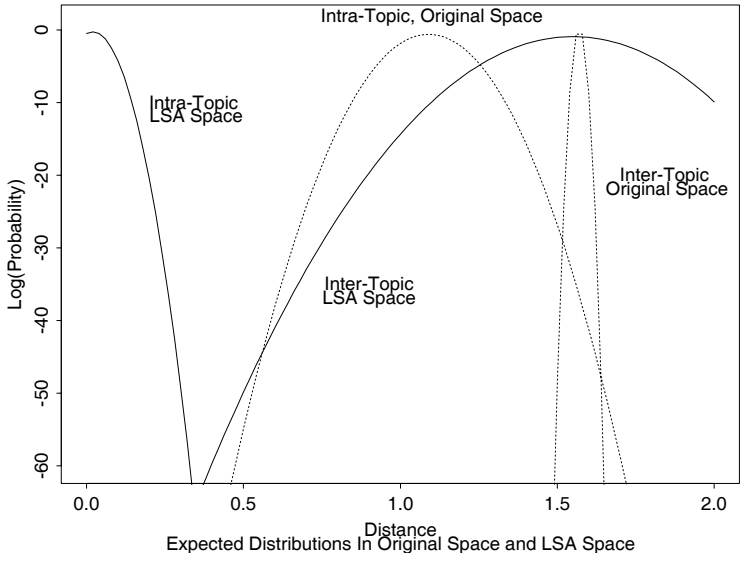

FIGURE 9.1
Improved Topic Separability in LSA Space.

the two documents were generated from the same topic and "inter-topic" otherwise. It can be seen that in the LSA space the average distance between inter-topic pairs stays about the same, while the average distance between intra-topic pairs is dramatically reduced. In addition, the intra-topic standard deviation also becomes substantially smaller. As a result, separability between intra- and inter-topic pairs is much better in the LSA space than in the original space. Interestingly, this holds despite a sharp increase in the inter-topic standard deviation, which bodes well for the general applicability of the method. Analogous observations can be made regarding the distance between words and/or between words and documents.

9.3 LSA Feature Space

In the continuous vector space $\mathcal{S}$ obtained above, each word $w_i \in \mathcal{V}$ is represented by the associated word vector of dimension R, $\bar{u}_i = u_i S$, and each document $d_j \in \mathcal{T}$ is represented by the associated document vector of dimension R, $\bar{v}_j = v_j S$. This opens up the opportunity to apply familiar clustering techniques in $\mathcal{S}$, as long as a distance measure consistent with the SVD formalism is defined on the vector space. Since the matrix W embodies, by construction, all structural associations

between words and documents, it follows that, for a given training corpus, $W\,W^{\,T}$ characterizes all co-occurrences between words, and $W^{\,T}\,W$ characterizes all co-occurrences between documents.

9.3.1 Word Clustering

Expanding $W\,W^{\,T}$ using the SVD expression (9.5), we obtain (henceforth ignoring the distinction between W and $\hat{W}$):

$$W\,W^{\,T} = U\,S^{\,2}\,U^{\,T}\,. \tag{9.6}$$

Since S is diagonal, a natural metric to consider for the "closeness" between words is therefore the cosine of the angle between $u_{\,i}S$ and $u_{\,j}S$:

$$K(w_i, w_j) = \cos(\bar{u}_i, \bar{u}_j) = \frac{u_i\,S^{\,2}\,u_j^{\,T}}{\|u_i S\|\,\|u_j S\|}\,, \tag{9.7}$$

for any $1 \leq i, j \leq M$. A value of $K(w_i, w_j) = 1$ means the two words always occur in the same semantic context, while a value of $K(w_i, w_j) < 1$ means the two words are used in increasingly different semantic contexts. While (9.7) does not define a *bona fide* distance measure in the space $\mathcal{S}$, it easy leads to one. For example, over the interval $[0, \pi]$, the measure:

$$\mathcal{D}(w_i, w_j) = \cos^{\,-1} K(w_i, w_j)\,, \tag{9.8}$$

readily satisfies the properties of a distance on $\mathcal{S}$. At this point, it is straightforward to proceed with the clustering of the word vectors $\bar{u}_{\,i}$, using any of a variety of algorithms (see, for instance, [1]). The outcome is a set of clusters $C_{\,k}$, $1 \leq k \leq K$, which can be thought of as revealing a particular layer of semantic knowledge in the space $\mathcal{S}$.

9.3.2 Word Cluster Example

To illustrate, we recall here the result of a word clustering experiment originally reported in [2]. A corpus of $N = 21,000$ documents was randomly selected from the WSJ portion of the NAB corpus. LSA training was then performed with an underlying vocabulary of $M = 23,000$ words, and the word vectors in the resulting LSA space were clustered into 500 disjoint clusters using a combination of K-means and bottom-up clustering (cf. [4]). Two representative examples of the clusters so obtained are shown in Figure 9.2.

In a marked difference with conventional class n-gram techniques (cf. [46]), these clusters comprise words with different part of speech. This is a direct consequence of the semantic nature of the derivation. Also, some obvious words seem to be missing from the clusters: for example, the singular noun "*drawing*" from cluster 1 and the present tense verb "*rule*" from cluster 2. This is an instance of *polysemy*: "*drawing*" and "*rule*" are more likely to appear in the training text with their alternative

Cluster 1
Andy, antique, antiques, art, artist, artist's, artists, artworks, auctioneers, Christie's, collector, drawings, gallery, Gogh, fetched, hysteria, masterpiece, museums, painter, painting, paintings, Picasso, Pollock, reproduction, Sotheby's, van, Vincent, Warhol
Cluster 2
appeal, appeals, attorney, attorney's, counts, court, court's, courts, condemned, convictions, criminal, decision, defend, defendant, dismisses, dismissed, hearing, here, indicted, indictment, indictments, judge, judicial, judiciary, jury, juries, lawsuit, leniency, overturned, plaintiffs, prosecute, prosecution, prosecutions, prosecutors, ruled, ruling, sentenced, sentencing, suing, suit, suits, witness

FIGURE 9.2
Word Cluster Example (After [2]).

meanings (as in "*drawing a conclusion*" and "*breaking a rule*," respectively), thus resulting in different cluster assignments. Finally, some words seem to contribute only marginally to the clusters: for example, "*hysteria*" from cluster 1 and "*here*" from cluster 2. These are the unavoidable outliers at the periphery of the clusters.

9.3.3 Document Clustering

Proceeding in a similar fashion at the document level, we obtain:

$$W^T W = V S^2 V^T, \tag{9.9}$$

which, for $1 \leq i, j \leq N$, leads to the same functional form as (9.7):

$$K(d_i, d_j) = \cos(\bar{v}_i, \bar{v}_j) = \frac{v_i S^2 v_j^T}{\|v_i S\| \, \|v_j S\|}. \tag{9.10}$$

We conclude that the distance (9.8) is equally valid for both word and document clustering.* The resulting set of clusters D_ℓ, $1 \leq \ell \leq L$, can be viewed as revealing another layer of semantic knowledge in the space $\mathcal{S}$.

*In fact, the measure (9.8) is precisely the one used in the study reported in Figure 9.1. Thus, the distances on the x-axis of Figure 9.1 are $\mathcal{D}(d_i, d_j)$ expressed in radians.

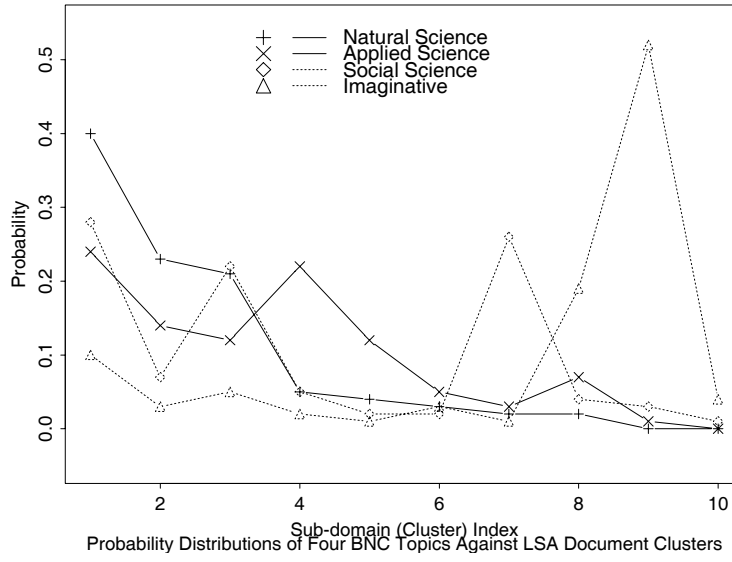

FIGURE 9.3
Document Cluster Example.

9.3.4 Document Cluster Example

An early document clustering experiment using the above measure was documented in [31]. This work was conducted on the British National Corpus (BNC), a heterogeneous corpus which contains a variety of hand-labelled topics. The LSA framework was used to partition BNC into distinct clusters, and the sub-domains so obtained were compared with the hand-labelled topics provided with the corpus. This comparison was conducted in an objective manner by evaluating (on a common test set) two different mixture trigram LMs: one built from the LSA sub-domains, and the other from the hand-labelled topics. As the perplexities obtained were very similar [31], it showed that the automatic partitioning performed using LSA was indeed semantically coherent.

Some evidence of this behavior is provided in Figure 9.3, which plots the distributions of four of the hand-labelled BNC topics against the ten document sub-domains automatically derived using LSA. Although it is clear that the data-driven sub-domains do not exactly match the hand-labeling, LSA document clustering in this example still seems reasonable. In particular, as one would expect, the distribution for the natural science topic is relatively close to the distribution for the applied science topic (cf. the two solid lines), but quite different from the two other topic distributions (in dashed lines). From that standpoint, the data-driven LSA clusters appear to adequately cover the semantic space.

9.4 Semantic Classification

To summarize, the latent semantic framework has a number of interesting properties, including: (i) a single vector representation for both words and documents in the same continuous vector space, (ii) an underlying topological structure reflecting semantic similarity, (iii) a well-motivated, natural metric to measure the distance between words and between documents in that space, and (iv) a relatively low dimensionality which makes clustering meaningful and practical. These properties can be exploited in several areas of spoken language processing. In this section, we address the most immediate domain of application, which follows directly from the previous clustering discussion: (data-driven) semantic classification [8, 9, 13, 17, 28].

9.4.1 Framework Extension

Semantic classification determines, for a given document, which one of several predefined topics the document is most closely aligned with. In contrast with the clustering setup discussed above, such document will not (normally) have been seen in the training corpus. Hence, we first need to extend the LSA framework accordingly. As it turns out, under relatively mild assumptions, finding a representation for a new document in the space $\mathcal{S}$ is straightforward.

Let us denote the new document by $\tilde{d}_p$, where the tilde symbol reflects the fact that $p > N$. It is obtained by constructing a feature vector containing, for each word in the underlying vocabulary, the weighted counts (9.3) with $j = p$. This vector $\tilde{d}_p$, as a column vector of dimension M, can be thought of as an additional column of the matrix W. Thus, provided the matrices U and S do not change, the SVD expansion (9.5) implies:

$$\tilde{d}_p = U\, S\, \tilde{v}_p^{\,T}\,, \tag{9.11}$$

where the R-dimensional vector $\tilde{v}_p^{\,T}$ acts as an additional column of the matrix $V^{\,T}$. This in turn leads to the definition:

$$\tilde{\tilde{v}}_p = \tilde{v}_p\, S = \tilde{d}_p^{\,T}\, U\,. \tag{9.12}$$

The vector $\tilde{\tilde{v}}_p$, indeed seen to be functionally similar to a document vector, corresponds to the representation of the new document in the space $\mathcal{S}$.

To convey the fact that it was not part of the SVD extraction, $\tilde{\tilde{v}}_p$ is referred to as a *pseudo document vector*. Recall that the (truncated) SVD provides, by definition, a parsimonious description of the linear space spanned by W. If the new document contains language patterns which are inconsistent with those extracted from W, the SVD expansion (9.5) will no longer apply. Similarly, if the addition of $\tilde{d}_p$ causes the major structural associations in W to shift in some substantial manner, the parsimonious description will become inadequate. Then U and S will no longer be valid, in which case it would be necessary to re-compute (9.5) to find a proper representation

for $\tilde{d}_p$.† If, on the other hand, the new document generally conforms to the rest of the corpus $\mathcal{T}$, then $\tilde{\tilde{v}}_p$ in (9.12) will be a reasonable representation for it.

Once the representation (9.12) is obtained, the "closeness" between the new document $\tilde{d}_p$ and any document cluster D_ℓ can then be expressed as $\mathcal{D}(\tilde{d}_p, D_\ell)$, calculated from (9.10) in the previous section.

9.4.2 Semantic Inference

This can be readily exploited in such command-and-control tasks as desktop user interface control [8] or automated call routing [13]. Suppose that each document cluster D_ℓ can be uniquely associated with a particular action in the task. Then the centroid of each cluster can be viewed as the *semantic anchor* of this action in the LSA space. An unknown word sequence (treated as a new "document") can thus be mapped onto an action by evaluating the distance (9.8) between that "document" and each semantic anchor. We refer to this approach as *semantic inference* [8, 9]. In contrast with usual inference engines (cf. [21]), semantic inference thus defined does not rely on formal behavioral principles extracted from a knowledge base. Instead, the domain knowledge is automatically encapsulated in the LSA space in a data-driven fashion.

To illustrate, consider an application with $N = 4$ actions (documents), each associated with a unique command: (i) "*what is the time*," (ii) "*what is the day*," (iii) "*what time is the meeting*," and (iv) "*cancel the meeting*." This simple example, with a vocabulary of only $M = 7$ words, is designed such that "*what*" and "*is*" always co-occur, "*the*" appears in all four commands, only (ii) and (iv) contain a unique word, and (i) is a proper subset of (iii). Constructing the (7×4) word-document matrix as above, and performing the SVD, we obtain the 2-dimensional space $\mathcal{S}$ shown in Figure 9.4, where is depicted the representation of each word and each command in the application.

The two words which each uniquely identify a command—"*day*" for (ii) and "*cancel*" for (iv)—each have a high coordinate on a different axis. Conversely, the word "*the*," which conveys no information about the identity of a command, is located at the origin. On the other hand, the semantic anchors for (ii) and (iv) fall "close" to the words which predict them best—"*day*" and "*cancel*", respectively. Similarly, the semantic anchors for (i) and (iii) fall in the vicinity of their meaningful components—"*what–is*" and "*time*" for (i) and "*time*" and "*meeting*" for (iii)—with the word "*time*," which occurs in both, indeed appearing "close" to both. Now suppose that a user says something outside of the training setup, such as "*when is the meeting*" rather than "*what time is the meeting*." This new word string (or variant) is represented in the space $\mathcal{S}$ by the hollow triangle in Figure 9.4, which is closest to

†For example, suppose training was carried out for a banking application involving the word "bank" taken in a financial context. Now suppose $\tilde{d}_p$ is germane to a fishing application, where "bank" is referred to in the context of a river or a lake. Clearly, the closeness of "bank" to, e.g., "money" and "account," would be irrelevant to $\tilde{d}_p$. Conversely, adding $\tilde{d}_p$ to W would likely cause such structural associations to shift substantially, and perhaps even disappear altogether.

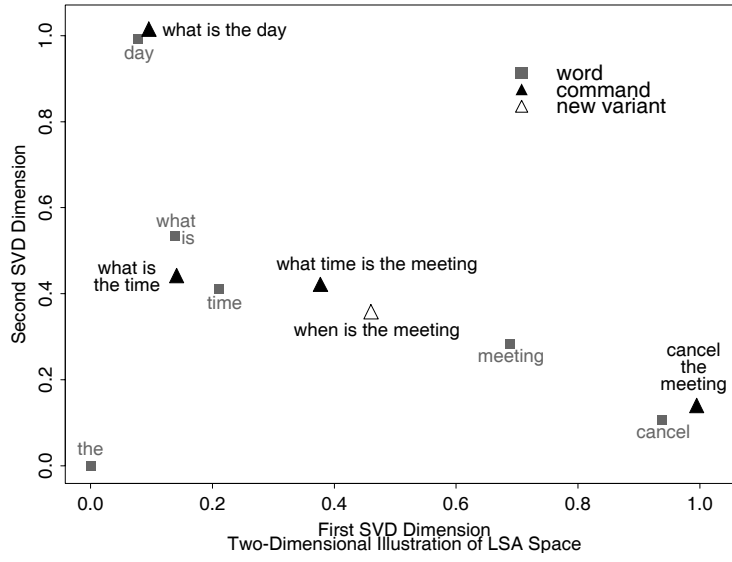

FIGURE 9.4
An Example of Semantic Inference for Command and Control ($R = 2$).

the representation of command (iii). Thus, the new variant appears most semantically related to (iii), and the correct action can be automatically inferred.

This can be thought of as a way to perform "bottom-up" natural language understanding. By replacing the traditional rule-based mapping between utterance and action by such data-driven classification, semantic inference makes it possible to relax some of the typical command-and-control interaction constraints. For example, it obviates the need to specify rigid language constructs through a domain-specific (and thus typically hand-crafted) finite state grammar. This is turn allows the end user more flexibility in expressing the desired command/query, which tends to reduce the associated cognitive load and thereby enhance user satisfaction [13].

9.4.3 Caveats

As an instance of the "bag-of-words" paradigm, LSA pays no attention to the order of words in sentences, which makes it ideally suited to capture large-span semantic relationships. By the same token, however, it is inherently unable to capitalize on the local (syntactic, pragmatic) constraints present in the language. For tasks such as call routing, which only needs to identify the broad topic of a message, this limitation is probably inconsequential. For general command and control tasks, however, it may be more deleterious.

Imagine two commands that differ only in the presence of the word "*not*" in a crucial place. The respective vector representations could conceivably be relatively close

in the LSA space, and yet have vastly different intended consequences. Worse yet, some commands may differ only through word order. Consider, for instance, the two MacOS 9 commands:

$$\begin{array}{l} \textit{change popup to window} \\ \textit{change window to popup} \end{array} \tag{9.13}$$

which are obviously impossible to disambiguate, since they are mapped onto the *exact same point* in LSA space.

As it turns out, it is possible to handle such cases through an extension of the basic LSA framework using word agglomeration. The idea is to move from words and documents to word n-tuples and n-tuple documents, where each word n-tuple is the agglomeration of n successive words, and each (n-tuple) document is now expressed in terms of all the word n-tuples it contains. Despite the resulting increase in computational complexity, this extension is practical in the context of semantic classification because of the relatively modest dimensions involved (as compared to large vocabulary recognition). Further details would be beyond the scope of this manuscript, but the reader is referred to [9] for a complete description.

9.5 N-gram+LSA Language Modeling

Another major area of application of the LSA framework is in statistical LM, where it can readily serve as a paradigm for semantically-driven span extension. Because of the limitation just discussed, however, it is best applied in conjunction with the standard n-gram approach. This section describes how this can be done.

9.5.1 LSA Component

Let w_q denote the word about to be predicted, and H_{q-1} the admissible LSA history (context) for this particular word. This notation translates a causality restriction of the context to $\tilde{d}_{q-1}$, the current document so far (i.e., up to word w_{q-1}). Thus, in general terms, the LSA LM probability is given by:

$$\Pr\left(w_q | H_{q-1}, \mathcal{S}\right) = \Pr\left(w_q | \tilde{d}_{q-1}\right), \tag{9.14}$$

where the conditioning on $\mathcal{S}$ reflects the fact that the probability depends on the particular vector space arising from the SVD representation. In this expression, $\Pr\left(w_q | \tilde{d}_{q-1}\right)$ is computed directly from the representations of w_q and $\tilde{d}_{q-1}$ in the space $\mathcal{S}$, i.e., it is inferred from the "closeness" between the associated word vector and (pseudo) document vector in $\mathcal{S}$. We therefore have to specify both the appropriate pseudo document representation and the relevant probability measure.

9.5.1.1 *Pseudo Document Representation*

To come up with a pseudo document representation, we leverage the results of Section 9.4.1, with some slight modifications due to the time-varying nature of the span considered. From (9.12), $\tilde{d}_{q-1}$ leads to the representation:

$$\tilde{\tilde{v}}_{q-1} = \tilde{v}_{q-1}\, S = \tilde{d}_{q-1}^{T}\, U\,. \tag{9.15}$$

As mentioned before, this pseudo vector representation is adequate under some consistency conditions on the general patterns present in the domain. The difference with Section 9.4.1 is that, as q increases, the content of the new document grows and the pseudo document vector moves around accordingly in the LSA space. Assuming the new document is semantically homogeneous, eventually we can expect the resulting trajectory to settle down in the vicinity of the document cluster corresponding to the closest semantic content.

Of course, here it is possible to take advantage of redundancies in time. Assume, without loss of generality, that word w_i is observed at time q. Then, $\tilde{d}_{q-1}$ and $\tilde{d}_q$ differ only in one coordinate, corresponding to the index i. Assume further that the training corpus $\mathcal{T}$ is large enough, so that the normalized entropy ε_i $(1 \leq i \leq M)$ does not change appreciably with the addition of each pseudo document. This makes it possible, from (9.3), to express $\tilde{d}_q$ as:

$$\tilde{d}_q = \frac{n_q - 1}{n_q}\,\tilde{d}_{q-1} \;+\; \frac{1-\varepsilon_i}{n_q}[0\ldots 1\ldots 0]^{T}\,, \tag{9.16}$$

where the "1" appears at coordinate i. This is turn implies, from (9.15):

$$\tilde{\tilde{v}}_q = \tilde{v}_q\, S = \frac{1}{n_q}\left[(n_q - 1)\,\tilde{\tilde{v}}_{q-1} \;+\; (1-\varepsilon_i)\,u_i\right]. \tag{9.17}$$

As a result, the pseudo document vector associated with the large-span context can be efficiently updated directly in the LSA space.

9.5.1.2 *LSA Probability*

To specify a suitable "closeness" measure, we now follow a reasoning similar to that of Section 9.3. Since, by construction, the matrix W embodies structural associations between words and documents, and, by definition, $W = USV^{T}$, a natural metric to consider for the "closeness" between word w_i and document d_j is the cosine of the angle between $u_i S^{1/2}$ and $v_j S^{1/2}$. Applying the same reasoning to pseudo documents, we arrive at:

$$K(w_q, \tilde{d}_{q-1}) = \cos(u_q S^{1/2}, \tilde{v}_{q-1} S^{1/2}) = \frac{u_q\, S\, \tilde{v}_{q-1}^{T}}{\|u_q S^{1/2}\|\,\|\tilde{v}_{q-1} S^{1/2}\|}\,, \tag{9.18}$$

for any q indexing a word in the text data. A value of $K(w_q, \tilde{d}_{q-1}) = 1$ means that $\tilde{d}_{q-1}$ is a strong semantic predictor of w_q, while a value of $K(w_q, \tilde{d}_{q-1}) < 1$

means that the history carries increasingly less information about the current word. Interestingly, (9.18) is functionally equivalent to (9.7) and (9.10), but involves scaling by $S^{1/2}$ instead of S. As before, the mapping (9.8) can be used to transform (9.18) into a real distance measure.

To enable the computation of $\Pr(w_q|\tilde{d}_{q-1})$, it remains to go from that distance measure to an actual probability measure. One possible solution is for the distance measure to induce a family of exponential distributions with pertinent marginality constraints. In practice, it may not be necessary to incur this degree of complexity. Considering that $\tilde{d}_{q-1}$ is only a partial document anyway, exactly what kind of distribution is induced is probably less consequential than ensuring that the pseudo document is properly scoped (cf. Section 9.5.3 below). Basically, all that is needed is a "reasonable" probability distribution to act as a proxy for the true (unknown) measure.

We therefore opt to use the empirical multivariate distribution constructed by allocating the total probability mass in proportion to the distances observed during training. In essence, this reduces the complexity to a simple histogram normalization, at the expense of introducing a potential "quantization-like" error. Of course, such error can be minimized through a variety of histogram smoothing techniques. Also note that the dynamic range of the distribution typically needs to be controlled by a parameter that is optimized empirically, e.g., by an exponent on the distance term, as discussed in [19].

Intuitively, $\Pr(w_q|\tilde{d}_{q-1})$ reflects the "relevance" of word w_q to the admissible history, as observed through $\tilde{d}_{q-1}$. As such, it will be highest for words whose meaning aligns most closely with the semantic fabric of $\tilde{d}_{q-1}$ (i.e., relevant "content" words), and lowest for words which do not convey any particular information about this fabric (e.g., "function" words like "*the*"). This behavior is exactly the opposite of that observed with the conventional n-gram formalism, which tends to assign higher probabilities to (frequent) function words than to (rarer) content words. Hence, the attractive synergy potential between the two paradigms.

9.5.2 Integration with N-grams

Exploiting this potential requires leveraging the benefits of both in a constructive manner. This kind of integration can occur in a number of ways, such as simple interpolation [19, 35], or within the maximum entropy framework [23, 42, 58]. Alternatively, under relatively mild assumptions, it is also possible to derive an integrated formulation directly from the expression for the overall LM probability. We start with the definition:

$$\Pr(w_q|H_{q-1}^{(n+l)}) = \Pr(w_q|H_{q-1}^{(n)}, H_{q-1}^{(l)}), \tag{9.19}$$

where H_{q-1} denotes, as before, some suitable admissible history for word w_q, and the superscripts $^{(n)}$, $^{(l)}$, and $^{(n+l)}$ refer to the n-gram component ($w_{q-1}w_{q-2}\cdots w_{q-n+1}$, with $n > 1$), the LSA component ($\tilde{d}_{q-1}$), and the integration thereof, re-

spectively. This expression can be rewritten as:

$$\Pr\left(w_q|H_{q-1}^{(n+l)}\right) = \frac{\Pr\left(w_q, H_{q-1}^{(l)}|H_{q-1}^{(n)}\right)}{\sum_{w_i\in\mathcal{V}} \Pr\left(w_i, H_{q-1}^{(l)}|H_{q-1}^{(n)}\right)}, \tag{9.20}$$

where the summation in the denominator extends over all words in $\mathcal{V}$. Expanding and re-arranging, the numerator of (9.20) is seen to be:

$$\begin{aligned}\Pr\left(w_q, H_{q-1}^{(l)}|H_{q-1}^{(n)}\right) &= \\ \Pr\left(w_q|H_{q-1}^{(n)}\right) &\cdot \Pr\left(H_{q-1}^{(l)}|w_q, H_{q-1}^{(n)}\right) = \\ \Pr\left(w_q|w_{q-1}w_{q-2}\ldots w_{q-n+1}\right)& \\ \cdot \Pr\left(\tilde{d}_{q-1}|w_q\, w_{q-1}w_{q-2}\ldots w_{q-n+1}\right)&.\end{aligned} \tag{9.21}$$

Now we make the assumption that the probability of the document history given the current word is not affected by the immediate context preceding it. This reflects the fact that, for a given word, different syntactic constructs (immediate context) can be used to carry the same meaning (document history). This is obviously reasonable for content words. How much it matters for function words is less clear [38], but we conjecture that if the document history is long enough, the semantic anchoring is sufficiently strong for the assumption to hold. As a result, the integrated probability becomes:

$$\begin{aligned}\Pr\left(w_q|H_{q-1}^{(n+l)}\right) &= \\ &\frac{\Pr\left(w_q|w_{q-1}w_{q-2}\ldots w_{q-n+1}\right)\Pr\left(\tilde{d}_{q-1}|w_q\right)}{\sum_{w_i\in\mathcal{V}} \Pr\left(w_i|w_{q-1}w_{q-2}\ldots w_{q-n+1}\right)\Pr\left(\tilde{d}_{q-1}|w_i\right)}.\end{aligned} \tag{9.22}$$

If $\Pr\left(\tilde{d}_{q-1}|w_q\right)$ is viewed as a prior probability on the current document history, then (9.22) simply translates the classical Bayesian estimation of the n-gram (local) probability using a prior distribution obtained from (global) LSA. The end result, in effect, is a modified n-gram LM incorporating large-span semantic information.
The dependence of (9.22) on the LSA probability calculated earlier can be expressed explicitly by using Bayes' rule to get $\Pr\left(\tilde{d}_{q-1}|w_q\right)$ in terms of $\Pr\left(w_q|\tilde{d}_{q-1}\right)$. Since the quantity $\Pr\left(\tilde{d}_{q-1}\right)$ vanishes from both numerator and denominator, we are left with:

$$\begin{aligned}\Pr\left(w_q|H_{q-1}^{(n+l)}\right) &= \\ &\frac{\Pr\left(w_q|w_{q-1}w_{q-2}\ldots w_{q-n+1}\right)\dfrac{\Pr\left(w_q|\tilde{d}_{q-1}\right)}{\Pr\left(w_q\right)}}{\sum_{w_i\in\mathcal{V}} \Pr\left(w_i|w_{q-1}w_{q-2}\ldots w_{q-n+1}\right)\dfrac{\Pr\left(w_i|\tilde{d}_{q-1}\right)}{\Pr(w_i)}},\end{aligned} \tag{9.23}$$

where $\Pr(w_q)$ is simply the standard unigram probability. Note that this expression is meaningful for any $n > 1$.[‡]

9.5.3 Context Scope Selection

In practice, expressions like (9.22)–(9.23) are often slightly modified so that a relative weight can be placed on each contribution (here, the n-gram and LSA probabilities). Usually, this is done via empirically determined weighting coefficients. In the present case, such weighting is motivated by the fact that in (9.22) the "prior" probability $\Pr(\tilde{d}_{q-1}|w_q)$ could change substantially as the current document unfolds. Thus, rather than using arbitrary weights, an alternative approach is to dynamically tailor the document history $\tilde{d}_{q-1}$ so that the n-gram and LSA contributions remain empirically balanced.

This approach, referred to as context scope selection, is more closely aligned with the LSA framework, because of the underlying change in behavior between training and recognition. During training, the scope is fixed to be the current document. During recognition, however, the concept of "current document" is ill-defined, because (i) its length grows with each new word, and (ii) it is not necessarily clear at which point completion occurs. As a result, a decision has to be made regarding what to consider "current," versus what to consider part of an earlier (presumably less relevant) document.

A straightforward solution is to limit the size of the history considered, so as to avoid relying on old, possibly obsolete fragments to construct the current context. Alternatively, to avoid making a hard decision on the size of the caching window, it is possible to assume an exponential decay in the relevance of the context [3]. In this solution, exponential forgetting is used to progressively discount older utterances. Assuming $0 < \lambda \leq 1$, this approach corresponds to modifying (9.17) as follows:

$$\tilde{\tilde{v}}_q = \frac{1}{n_q} \left[\lambda \left(n_q - 1 \right) \tilde{\tilde{v}}_{q-1} \; + \; \left(1 - \varepsilon_i \right) u_i \right], \tag{9.24}$$

where the parameter λ is chosen according to the expected heterogeneity of the session.

In terms of computational effort, the (on-line) cost incurred during recognition comprises: (i) the construction of the pseudo document representation in $\mathcal{S}$, as generally done via (9.24); (ii) the computation of the LSA probability $\Pr(w_q|\tilde{d}_{q-1})$ in (9.14); and (iii) the integration proper, in (9.23). It can be shown (cf. [3, 4]) that the total cost of these operations, per word and pseudo document, is $\mathcal{O}(R^2)$. This is obviously more expensive than the usual table look-up required in conventional n-gram LM. Yet, for typical values of R, the resulting overhead is, arguably, quite modest. This allows hybrid n-gram+LSA LM to be brought to bear in early stages of the search [3].

[‡]Moreover, the assumption that $n > 1$ is without loss of generality. When $n = 1$, the right hand side of (9.23) degenerates to the LSA probability alone, since the n-gram history becomes null. But the integrated history also degenerates to the LSA history alone, effectively reducing (9.19) to (9.14).

9.6 Smoothing

Since the derivation of (9.23) does not depend on a particular form of the LSA probability, it is possible to take advantage of the additional layer(s) of knowledge uncovered earlier through word (in Section 9.3.1) and document (in Section 9.3.3) clustering. Basically, we can expect words and/or documents related to the current document to contribute with more synergy, and unrelated words and/or documents to be better discounted. Clustering therefore provides a convenient smoothing mechanism in the LSA space [2, 3].

9.6.1 Word Smoothing

Using the set of word clusters C_k, $1 \leq k \leq K$, produced in Section 9.3.1 leads to word-based smoothing. In this case, we expand (9.14) as follows:

$$\Pr\left(w_q|\tilde{d}_{q-1}\right) = \sum_{k=1}^{K} \Pr\left(w_q|C_k\right) \Pr\left(C_k|\tilde{d}_{q-1}\right), \tag{9.25}$$

which carries over to (9.23) in a straightforward manner. In (9.25), the probability $\Pr(C_k|\tilde{d}_{q-1})$ is qualitatively similar to (9.14) and can therefore be obtained with the help of (9.18), by simply replacing the representation of the word w_q by that of the centroid of word cluster C_k. In contrast, the probability $\Pr(w_q|C_k)$ depends on the "closeness" of w_q relative to this (word) centroid. To derive it, we therefore have to rely on the empirical multivariate distribution induced not by the distance obtained from (9.18), but by that obtained from the measure (9.7) mentioned in Section 9.3.1. Note that a distinct distribution can be inferred on each of the clusters C_k, thus allowing us to compute all quantities $\Pr\left(w_i|C_k\right)$ for $1 \leq i \leq M$ and $1 \leq k \leq K$.
The behavior of the model (9.25) depends on the number of word clusters defined in the space $\mathcal{S}$. Two special cases arise at the extremes of the cluster range. If there are as many classes as words in the vocabulary ($K = M$), then with the convention that $P(w_i|C_j) = \delta_{ij}$, (9.25) simply reduces to (9.14). No smoothing is introduced, so the predictive power of the model stays the same as before. Conversely, if all the words are in a single class ($K = 1$), the model becomes maximally smooth: the influence of specific semantic events disappears, leaving only a residual vocabulary effect to take into account. The effect on predictive power is, accordingly, limited. Between these two extremes, as smoothness gradually increases, it is reasonable to postulate that predictive power goes through a peak.
The intuition behind this conjecture is as follows. Generally speaking, as the number of word classes C_k increases, the contribution of $\Pr(w_q|C_k)$ tends to increase, because the clusters become more and more semantically meaningful. By the same token, however, the contribution of $\Pr(C_k|\tilde{d}_{q-1})$ for a given $\tilde{d}_{q-1}$ tends to decrease, because the clusters eventually become too specific and fail to reflect the overall semantic fabric of $\tilde{d}_{q-1}$. Thus, there must exist a cluster set size where the degree

of smoothing (and therefore the associated predictive power) is optimal for the task considered. This has indeed been verified experimentally, cf. [2].

9.6.2 Document Smoothing

Exploiting instead the set of document clusters D_ℓ, $1 \leq \ell \leq L$, produced in Section 9.3.3 leads to document-based smoothing. The expansion is similar:

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{\ell=1}^{L} \Pr(w_q|D_\ell) \Pr(D_\ell|\tilde{d}_{q-1}), \tag{9.26}$$

with (document) clusters D_ℓ now replacing the (word) clusters C_k. This time, it is the probability $\Pr(w_q|D_\ell)$ which is qualitatively similar to (9.14), and can therefore be obtained with the help of (9.18). As for the probability $\Pr(D_\ell|\tilde{d}_{q-1})$, it depends on the "closeness" of $\tilde{d}_{q-1}$ relative to the centroid of document cluster D_ℓ. Thus, it can be obtained through the empirical multivariate distribution induced by the distance derived from (9.10) in Section 9.3.3.

Again, the behavior of the model (9.26) depends on the number of document clusters defined in the space $\mathcal{S}$. Compared to (9.25), however, (9.26) is more difficult to interpret at the extremes of the cluster range (i.e., $L = 1$ and $L = N$). If $L = N$, for example, (9.26) does not reduce to (9.14), because $\tilde{d}_{q-1}$ has not been seen in the training data, and therefore cannot be identified with any of the existing clusters. Similarly, the fact that all the documents are in a single cluster ($L = 1$) does not imply the degree of degenerateness observed previously, because the cluster itself is strongly indicative of the general discourse domain (which was not generally true of the "vocabulary cluster" above). Hence, depending on the size and structure of the corpus, the model may well be important to capture general discourse effects.

To see that, we apply $L = 1$ in (9.26), to obtain in (9.23):

$$\Pr(w_q|H_{q-1}^{(n+l)}) = \frac{\Pr(w_q|w_{q-1}w_{q-2}\ldots w_{q-n+1}) \dfrac{\Pr(w_q|D_1)}{\Pr(w_q)}}{\displaystyle\sum_{w_i \in \mathcal{V}} \Pr(w_i|w_{q-1}w_{q-2}\ldots w_{q-n+1}) \frac{\Pr(w_i|D_1)}{\Pr(w_i)}}, \tag{9.27}$$

since the quantity $\Pr(D_1|\tilde{d}_{q-1})$ vanishes from both numerator and denominator. In this expression D_1 refers to the single document cluster encompassing all documents in the LSA space. In case the corpus is fairly homogeneous, D_1 will be a more reliable representation of the underlying fabric of the domain than $\tilde{d}_{q-1}$, and therefore act as a robust proxy for the context observed. Interestingly, (9.27) amounts to estimating a "correction" factor for each word, which depends only on the overall topic of the collection. This is similar to what is done in the cache approach to LM adaptation (see, for example, [18, 41]), except that, here, all words are treated as though they were already in the cache.

More generally, as the number of document classes D_ℓ increases, the contribution of $\Pr(w_q|D_\ell)$ tends to increase, to the extent that a more homogeneous topic boosts the effects of any related content words. On the other hand, the contribution of $\Pr(D_\ell|\tilde{d}_{q-1})$ tends to decrease, because the clusters represent more and more specific topics, which increases the chance that the pseudo document $\tilde{d}_{q-1}$ becomes an outlier. Thus, again there exists a cluster set size where the degree of smoothing is optimal for the task considered (cf. [2]).

9.6.3 Joint Smoothing

Finally, an expression analogous to (9.25) and (9.26) can also be derived to take advantage of both word and document clusters. This leads to a mixture probability specified by:

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{k=1}^{K}\sum_{\ell=1}^{L} \Pr(w_q|C_k, D_\ell)\Pr(C_k, D_\ell|\tilde{d}_{q-1}), \tag{9.28}$$

which, for tractability, can be approximated as:

$$\Pr(w_q|\tilde{d}_{q-1}) = \sum_{k=1}^{K}\sum_{\ell=1}^{L} \Pr(w_q|C_k)\Pr(C_k|D_\ell)\Pr(D_\ell|\tilde{d}_{q-1}). \tag{9.29}$$

In this expression, the clusters C_k and D_ℓ are as previously, as are the quantities $\Pr(w_q|C_k)$ and $\Pr(D_\ell|\tilde{d}_{q-1})$. As for the probability $\Pr(C_k|D_\ell)$, it is qualitatively similar to (9.14), and can therefore be obtained accordingly.

To summarize, any of the expressions (9.14), (9.25), (9.26), or (9.29) can be used to compute (9.23). This results in four families of hybrid n-gram+LSA LMs. Associated with these different families are various trade-offs to become apparent below.

9.7 Experiments

The purpose of this section is to illustrate the behavior of hybrid n-gram+LSA modeling on a large vocabulary recognition task.[§] The general domain considered was business news, as reflected in the WSJ portion of the NAB corpus. This was convenient for comparison purposes since conventional n-gram LMs are readily available, trained on exactly the same data [40].

[§]The reader is referred to [4] for additional results in this application, and to [9] for experiments involving semantic inference.

9.7.1 Experimental Conditions

The text corpus $\mathcal{T}$ used in this training was composed of about $N = 87,000$ documents spanning the years 1987 to 1989, comprising approximately 42 million words. The vocabulary $\mathcal{V}$ was constructed by taking the 20,000 most frequent words of the NAB corpus, augmented by some words from an earlier release of the WSJ corpus, for a total of $M = 23,000$ words. The test set consisted of a 1992 test corpus of 496 sentences uttered by 12 native speakers of English. In all experiments, acoustic training was performed using 7,200 sentences of data uttered by 84 speakers (WSJ0 SI-84). On the above test data, our baseline speaker-independent, continuous speech recognition system (described in detail in [3]) produced reference error rates of 16.7% and 11.8% across the 12 speakers considered, using the standard (WSJ0) bigram and trigram LMs, respectively.

After feature extraction using (9.3), we performed the singular value decomposition of the matrix of co-occurrences between words and documents using the single vector Lanczos method [10]. Over the course of this decomposition, we experimented with different numbers of singular values retained, and found that $R = 125$ seemed to achieve an adequate balance between reconstruction error—minimizing s_{R+1}, the largest singular value not retained—and noise suppression—minimizing the ratio between order-R and order-$(R+1)$ traces $\sum_i s_i$. This led to a vector space $\mathcal{S}$ of dimension 125.

Following Section 9.5, we then used this LSA space to construct the (unsmoothed) LSA model (9.14). We also constructed the various clustered LSA models presented in Section 9.6, to implement word smoothing based on (9.25), document smoothing based on (9.26), and joint smoothing based on (9.29). We experimented with different values for the number of word and/or document clusters (cf. [2]), and ended up using $K = 100$ word clusters and $L = 1$ document cluster. Finally, using (9.23), we combined each of these models with either the standard WSJ0 bigram or the standard WSJ0 trigram. The resulting hybrid n-gram+LSA LMs, dubbed bi-LSA and tri-LSA models, respectively, were then used in lieu of the standard WSJ0 bigram and trigram models.

9.7.2 Experimental Results

A summary of the results is provided in Table 9.1, in terms of both absolute word error rate (WER) numbers and WER reduction observed (in angle brackets). Without smoothing, the bi-LSA LM leads to a 14% WER reduction compared to the standard bigram. The corresponding tri-LSA LM leads to a somewhat smaller (9%) relative improvement compared to the standard trigram. With smoothing, the improvement brought about by the LSA component is more marked: up to 23% in the smoothed bi-LSA case, and up to 16% in the smoothed tri-LSA case. Such results show that the hybrid n-gram+LSA approach is a promising avenue for incorporating large-span semantic information into n-gram modeling.

The qualitative behavior of the two n-gram+LSA LMs appears to be quite similar. Quantitatively, the average reduction achieved by tri-LSA is about 30% less than that

TABLE 9.1
Word Error Rate (WER) Results Using Hybrid Bi-LSA and Tri-LSA Models.

Word Error Rate <WER Reduction>	Bigram $n = 2$	Trigram $n = 3$
Conventional n-Gram	16.7 %	11.8 %
Hybrid, No Smoothing	14.4 % <14 %>	10.7 % < 9 %>
Hybrid, Document Smoothing	13.4 % <20 %>	10.4 % <12 %>
Hybrid, Word Smoothing	12.9 % <23 %>	9.9 % <16 %>
Hybrid, Joint Smoothing	13.0 % <22 %>	9.9 % <16 %>

achieved by bi-LSA. This is most likely related to the greater predictive power of the trigram compared to the bigram, which makes the LSA contribution of the hybrid LM comparatively smaller. This is consistent with the fact that the latent semantic information delivered by the LSA component would (eventually) be subsumed by an n-gram with a large enough n. Interestingly, in both cases the average WER reduction is far from constant across individual sessions, reflecting the varying role played by global semantic constraints from one set of spoken utterances to another.

9.7.3 Context Scope Selection

It is important to emphasize that the recognition task chosen above represents a severe test of the LSA component of the hybrid LM. By design, the test corpus is constructed with no more than three or four consecutive sentences extracted from a single article. Overall, it comprises 140 distinct document fragments, which means that each speaker speaks, on the average, about 12 different "mini-documents." As a result, the context effectively changes every 60 words or so, which makes it somewhat challenging to build a very accurate pseudo document representation. This is a situation where it is critical for the LSA component to appropriately forget the context as it unfolds, to avoid relying on an obsolete representation. To obtain the results of Table 9.1, we used the exponential forgetting setup of (9.24) with a value $\lambda = 0.975$.[¶]

In order to assess the influence of this selection, we also performed recognition with different values of the parameter λ ranging from $\lambda = 1$ to $\lambda = 0.95$, in decrements of 0.01. Recall from Section 9.5 that the value $\lambda = 1$ corresponds to an unbounded context (as would be appropriate for a very homogeneous session), while decreasing values of λ correspond to increasingly more restrictive contexts (as required for a more heterogeneous session). Hence, the gap between λ and 1 tracks the expected heterogeneity of the session.

Table 9.2 presents the corresponding recognition results, in the case of the best bi-LSA framework (i.e., with word smoothing). It can be seen that, with no forgetting,

[¶] To fix ideas, this means that the word which occurred 60 words ago is discounted through a weight of about 0.2.

TABLE 9.2
Influence of Context Scope Selection on Word Error Rate.

Word Error Rate <WER Reduction>	Bi-LSA with Word Smoothing
$\lambda = 1.0$	14.5 % <13 %>
$\lambda = 0.99$	13.6 % <18 %>
$\lambda = 0.98$	13.2 % <21 %>
$\lambda = 0.975$	12.9 % <23 %>
$\lambda = 0.97$	13.0 % <22 %>
$\lambda = 0.96$	13.1 % <22 %>
$\lambda = 0.95$	13.5 % <19 %>

the overall performance is substantially less than the comparable one observed in Table 9.1 (13% compared to 23% WER reduction). This is consistent with the characteristics of the task, and underscores the role of discounting as a suitable counterbalance to frequent context changes. Performance rapidly improves as λ decreases from $\lambda = 1$ to $\lambda = 0.97$, presumably because the pseudo document representation gets less and less contaminated with obsolete data. If forgetting becomes too aggressive, however, the performance starts degrading, as the effective context no longer has an equivalent length which is sufficient for the task at hand. Here, this happens for $\lambda < 0.97$.

9.8 Inherent Trade-Offs

In the previous section, both LSA and n-gram components of the hybrid LM were trained on exactly the same data. This is not a requirement, however, which raises the question of how critical the selection of the LSA training data is to the performance of the recognizer. This is particularly interesting since LSA is known to be weaker on heterogeneous corpora (see, for example, [31]).

9.8.1 Cross-Domain Training

To ascertain the matter, we went back to an LSA component involving the original, unsmoothed model (9.14). We kept the same underlying vocabulary $\mathcal{V}$, left the bigram component unchanged, and repeated the LSA training on non-WSJ data from the same general period. Three corpora of increasing size were considered, all corresponding to Associated Press (AP) data: (i) $\mathcal{T}_1$, composed of $N_1 = 84,000$ documents from 1989, comprising approximately 44 million words; (ii) $\mathcal{T}_2$, composed of $N_2 = 155,000$ documents from 1988 and 1989, comprising approximately 80 million words; and (iii) $\mathcal{T}_3$, composed of $N_3 = 224,000$ documents from 1988-1990,

TABLE 9.3
Model Sensitivity to LSA Training Data.

Word Error Rate <WER Reduction>	Bi-LSA with No Smoothing
$\mathcal{T}_1$: $N_1 = 84,000$	16.3 % <2 %>
$\mathcal{T}_2$: $N_2 = 155,000$	16.1 % <3 %>
$\mathcal{T}_3$: $N_3 = 224,000$	16.0 % <4 %>

comprising approximately 117 million words. In each case LSA training proceeded as described in Section 9.2. The results are reported in Table 9.3.

Two things are immediately apparent. First, the performance improvement in all cases is much smaller than the 14% reduction observed in Table 9.1. on the average, the hybrid model trained on AP data is about four times less effective than that trained on WSJ data. This suggests a relatively high LSA sensitivity to the domain considered. To put this observation in perspective, recall that: (i) by definition, a domain is characterized by content words; and (ii) LSA inherently relies on content words, since, in contrast with n-grams, it cannot take advantage of the structural aspects of the sentence. It therefore makes sense to expect a higher sensitivity for the LSA component than for the usual n-gram.

Second, the overall performance does not improve appreciably with more training data, a fact already observed in [2] using a perplexity measure. Larger training set sizes notwithstanding, LSA still detects a substantial mismatch between AP and WSJ data from the same general period. This supports the conjecture that LSA is sensitive not just to the general training domain, but also to the particular style of composition, as might be reflected, for example, in the choice of content words and/or word co-occurrences. On the positive side, this bodes well for rapid adaptation to cross-domain data, provided a suitable adaptation framework can be derived.

9.8.2 Discussion

The fact that the hybrid n-gram+LSA approach is sensitive to composition style underscores the relatively narrow semantic specificity of the LSA paradigm. While n-grams also suffer from any mismatch between training and recognition, LSA leads to a potentially more severe exposure because the space $\mathcal{S}$ reflects even less of the pragmatic characteristics for the task considered. Perhaps what is required is to explicitly include an "authorship style" component into the LSA framework. [‖] In any event, one has to be cognizant of this intrinsic limitation, and mitigate it through careful attention to the expected domain of use.

[‖] In [48], for example, it has been suggested to define an $M \times M$ stochastic matrix (a matrix with non-negative entries and row sums equal to 1) to account for the way style modifies the frequency of words. This solution, however, makes the assumption—not always valid—that this influence is independent of the underlying subject matter.

Another caveat is the fact that LSA is inherently more adept at handling content words than function words. As is well-known, a substantial proportion of speech recognition errors come from function words, because of their tendency to be shorter, not well articulated, and acoustically confusable. In general, LSA's contribution to fixing such problems will be limited. This suggests that, even within a well-specified domain, syntactically-driven span extension techniques may be a necessary complement to the hybrid approach.

On that subject, note from Section 9.5 that the integrated history (9.19) could easily be modified to reflect a headword-based n-gram as opposed to a conventional n-gram history, without invalidating the derivation of (9.23). Thus, there is no theoretical barrier to the integration of latent semantic information with structured LMs such as described in [15, 36]. Similarly, there is no reason why the LSA paradigm could not be used in conjunction with the integrative approaches of the kind proposed in [54, 58], or even within the cache adaptive framework [18, 41].

9.9 Conclusion

Statistical n-grams are by nature limited to the capture of linguistic phenomena spanning at most n words. This chapter has focused on a semantically-driven span extension framework based on the LSA paradigm, in which hidden semantic redundancies are tracked across semantically homogeneous documents. This approach leads to a (continuous) vector representation of each (discrete) word and document in a space of relatively modest dimension, in which suitable metrics can be defined for word-document, word-word, and document-document comparisons. As well-known clustering algorithms can then be applied efficiently, this makes it possible to uncover, in a data-driven fashion, multiple parallel layers of semantic knowledge with variable granularity.

An important property of this vector representation is that it reflects the major semantic associations in the training corpus, as determined by the overall pattern of the language, as opposed to specific word sequences or grammatical constructs. LMs constructed from the LSA framework are therefore well suited to complement conventional n-grams. Harnessing this synergy is a matter of deriving an integrative formulation to combine the two paradigms. By taking advantage of the various kinds of smoothing available, several families of hybrid n-gram+LSA models can be obtained. The resulting LMs substantially outperform the associated standard n-grams on a subset of the NAB News corpus.

Such results notwithstanding, hybrid n-gram+LSA modeling also faces some intrinsic limitations. For example, LSA shows marked sensitivity to both the training domain and the style of composition. While cross-domain adaptation may ultimately alleviate this problem, an appropriate LSA adaptation framework will have to be derived for this purpose (for some recent progress on that front, see [6]). More gen-

erally, semantically-driven span extensions like the one proposed here run the risk of lackluster improvement when it comes to function word recognition. This underscores the need for an all-encompassing strategy involving syntactically motivated approaches as well.

References

[1] J.R. Bellegarda, *Context-Dependent Vector Clustering for Speech Recognition*, Chapter 6 in Automatic Speech and Speaker Recognition: Advanced Topics, C.-H. Lee, F.K. Soong, and K.K. Paliwal (Eds.), Kluwer Academic Publishers, NY, pp. 133–157, March 1996.

[2] J.R. Bellegarda, *A Multi-Span Language Modeling Framework for Large Vocabulary Speech Recognition*, IEEE Trans. Speech Audio Proc., Vol. 6, No. 5, pp. 456–467, September 1998.

[3] J.R. Bellegarda, *Large Vocabulary Speech Recognition With Multi-Span Statistical Language Models*, IEEE Trans. Speech Audio Proc., Vol. 8, No. 1, pp. 76–84, January 2000.

[4] J.R. Bellegarda, *Exploiting Latent Semantic Information in Statistical Language Modeling*, Proc. IEEE, Spec. Issue Speech Recog. Understanding, B.H. Juang and S. Furui (Eds.), Vol. 88, No. 8, pp. 1279–1296, August 2000.

[5] J.R. Bellegarda, *Robustness in Statistical Language Modeling: Review and Perspectives*, Chapter 4 in Robustness in Language and Speech Technology, J.C. Junqua and G.J.M. van Noord (Eds.), Kluwer Academic Publishers, Dortrecht, The Netherlands, pp. 101–121, February 2001.

[6] J.R. Bellegarda, *Fast Update of Latent Semantic Spaces Using a Linear Transform Framework*, in Proc. 2002 Int. Conf. Acoust., Speech, Sig. Proc., Orlando, FL, May 2002.

[7] J.R. Bellegarda, J.W. Butzberger, Y.L. Chow, N.B. Coccaro, and D. Naik, *A Novel Word Clustering Algorithm Based on Latent Semantic Analysis*, in Proc. 1996 Int. Conf. Acoust., Speech, Sig. Proc., Atlanta, GA, pp. I172–I175, May 1996.

[8] J.R. Bellegarda and K.E.A. Silverman, *Toward Unconstrained Command and Control: Data-Driven Semantic Inference*, in Proc. Int. Conf. Spoken Language Proc., Beijing, China, pp. I258–I261, October 2000.

[9] J.R. Bellegarda and K.E.A. Silverman, *Natural Language Spoken Interface Control Using Data-Driven Semantic Inference*, IEEE Trans. Speech Audio Proc., in press.

[10] M.W. Berry, *Large–Scale Sparse Singular Value Computations*, Int. J. Supercomp. Appl., Vol. 6, No. 1, pp. 13–49, 1992.

[11] M.W. Berry, S.T. Dumais, and G.W. O'Brien, *Using Linear Algebra for Intelligent Information retrieval*, SIAM Review, Vol. 37, No. 4, pp. 573–595, 1995.

[12] M. Berry and A. Sameh, *An Overview of Parallel Algorithms for the Singular Value and Dense Symmetric Eigenvalue Problems*, J. Computational Applied Math., Vol. 27, pp. 191–213, 1989.

[13] B. Carpenter and J. Chu–Carroll, *Natural Language Call Routing: A Robust, Self–Organized Approach*, in Proc. Int. Conf. Spoken Language Proc., Sydney, Australia, pp. 2059–2062, December 1998.

[14] C. Chelba, D. Engle, F. Jelinek, V. Jimenez, S. Khudanpur, L. Mangu, H. Printz, E.S. Ristad, R. Rosenfeld, A. Stolcke and D. Wu, *Structure and Performance of a Dependency Language Model*, in Proc. Fifth Euro. Conf. Speech Comm. Technol., Rhodes, Greece, Vol. 5, pp. 2775–2778, September 1997.

[15] C. Chelba and F. Jelinek, *Recognition Performance of a Structured Language Model*, in Proc. Sixth Euro. Conf. Speech Comm. Technol., Budapest, Hungary, Vol. 4, pp. 1567–1570, September 1999.

[16] S. Chen, *Building Probabilistic Models for Natural Language*, Ph.D. Thesis, Harvard University, Cambridge, MA, 1996.

[17] J. Chu–Carroll and B. Carpenter, *Dialog Management in Vector–Based Call Routing*, in Proc. Conf. Assoc. Comput. Linguistics ACL/COLING, Montreal, Canada, pp. 256–262, 1998.

[18] P.R. Clarkson and A.J. Robinson, *Language Model Adaptation Using Mixtures and an Exponentially Decaying Cache*, in Proc. 1997 Int. Conf. Acoust., Speech, Signal Proc., Munich, Germany, Vol. 1, pp. 799–802, May 1997.

[19] N. Coccaro and D. Jurafsky, *Towards Better Integration of Semantic Predictors in Statistical Language Modeling*, in Proc. Int. Conf. Spoken Language Proc., Sydney, Australia, pp. 2403–2406, December 1998.

[20] J.K. Cullum and R.A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations – Vol. 1 Theory*, Chapter 5: Real Rectangular Matrices, Brickhauser, Boston, MA, 1985.

[21] R. De Mori, *Recognizing and Using Knowledge Structures in Dialog Systems*, in Proc. Aut. Speech Recog. Understanding Workshop, Keystone, CO, pp. 297–306, December 1999.

[22] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, *Indexing by Latent Semantic Analysis*, J. Am. Soc. Inform. Science, Vol. 41, pp. 391–407, 1990.

[23] S. Della Pietra, V. Della Pietra, R. Mercer, and S. Roukos, *Adaptive Language Model Estimation Using Minimum Discrimination Estimation*, in Proc. 1992 Int. Conf. Acoust., Speech, Signal Processing, San Francisco, CA, Vol. I, pp. 633–636, April 1992.

[24] S.T. Dumais, *Improving the Retrieval of Information from External Sources*, Behavior Res. Methods, Instrum., Computers, Vol. 23, No. 2, pp. 229–236, 1991.

[25] S.T. Dumais, *Latent Semantic Indexing (LSI) and TREC–2*, in Proc. Second Text Retrieval Conference (TREC–2), D. Harman (Ed.), NIST Pub. 500–215, pp. 105–116, 1994.

[26] M. Federico and R. De Mori, *Language Modeling*, Chapter 7 in Spoken Dialogues with Computers, R. De Mori (Ed.), Academic Press, London, UK, pp. 199–230, 1998.

[27] P.W. Foltz and S.T. Dumais, *Personalized Information Delivery: An Analysis of Information Filtering Methods*, Commun. ACM, Vol. 35, No. 12, pp. 51–60, 1992.

[28] P.N. Garner, *On Topic Identification and Dialogue Move Recognition*, Computer Speech and Language, Vol. 11, No. 4, pp. 275–306, 1997.

[29] D. Gildea and T. Hofmann, *Topic–Based Language Modeling Using EM*, in Proc. Sixth Euro. Conf. Speech Comm. Technol., Budapest, Hungary, Vol. 5, pp. 2167–2170, September 1999.

[30] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins, Baltimore, MD, Second Ed., 1989.

[31] Y. Gotoh and S. Renals, *Document Space Models Using Latent Semantic Analysis*, in Proc. Fifth Euro. Conf. Speech Comm. Technol., Rhodes, Greece, Vol. 3, pp. 1443–1448, September 1997.

[32] T. Hofmann, *Probabilistic Latent Semantic Analysis*, in Proc. Fifteenth Conf. Uncertainty in AI, Stockholm, Sweden, July 1999.

[33] T. Hofmann, *Probabilistic Topic Maps: Navigating Through Large Text Collections*, in Lecture Notes Comp. Science., No. 1642, pp. 161–172, Springer–Verlag, Heidelberg, Germany, July 1999.

[34] R. Iyer and M. Ostendorf, *Modeling Long Distance Dependencies in Language: Topic Mixtures Versus Dynamic Cache Models*, IEEE Trans. Speech Audio Proc., Vol. 7, No. 1, January 1999.

[35] F. Jelinek, *Self–Organized Language Modeling for Speech Recognition*, in Readings in Speech Recognition, A. Waibel and K.F. Lee (Eds.), Morgan Kaufmann Publishers, pp. 450–506, 1990.

[36] F. Jelinek and C. Chelba, *Putting Language into Language Modeling*, in Proc. Sixth Euro. Conf. Speech Comm. Technol., Budapest, Hungary, Vol. 1, pp. KN1–KN5, September 1999.

[37] D. Jurafsky, C. Wooters, J. Segal, A. Stolcke, E. Fosler, G. Tajchman, and N. Morgan, *Using a Stochastic Context–Free Grammar as a Language Model for Speech Recognition*, in Proc. 1995 Int. Conf. Acoust., Speech, Signal Proc., Detroit, MI, Vol. I, pp. 189–192, May 1995.

[38] S. Khudanpur, *Putting Language Back into Language Modeling*, presented at Workshop–2000 Spoken Lang. Reco. Understanding, Summit, NJ, February 2000.

[39] R. Kneser, *Statistical Language Modeling Using a Variable Context*, in Proc. Int. Conf. Spoken Language Proc., pp. 494–497, Philadelphia, PA, October 1996.

[40] F. Kubala, J.R. Bellegarda, J.R. Cohen, D. Pallett, D.B. Paul, M. Phillips, R. Rajasekaran, F. Richardson, M. Riley, R. Rosenfeld, R. Roth, and M. Weintraub, *The Hub and Spoke Paradigm for CSR Evaluation*, in Proc. ARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, pp. 40–44, March 1994.

[41] R. Kuhn and R. De Mori, *A Cache-based Natural Language Method for Speech Recognition*, IEEE Trans. Pattern Anal. Mach. Intel., Vol. PAMI–12, No. 6, pp. 570–582, June 1990.

[42] J.D. Lafferty and B. Suhm, *Cluster Expansion and Iterative Scaling for Maximum Entropy Language Models*, in Maximum Entropy and Bayesian Methods, K. Hanson and R. Silver (Eds.), Kluwer Academic Publishers, Norwell, MA, 1995.

[43] T.K. Landauer and S.T. Dumais, *Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge*, in Psychological Review, Vol. 104, No. 2, pp. 211–240, 1997.

[44] T.K. Landauer, D. Laham, B. Rehder, and M.E. Schreiner, *How Well Can Passage Meaning Be Derived Without Using Word Order: A Comparison of Latent Semantic Analysis and Humans*, in Proc. Conf. Cognit. Science Soc., Mahwah, NJ, pp. 412–417, 1997.

[45] R. Lau, R. Rosenfeld, and S. Roukos, *Trigger–Based Language Models: A Maximum Entropy Approach*, in Proc. 1993 Int. Conf. Acoust., Speech, Signal Proc., Minneapolis, MN, pp. II45–48, May 1993.

[46] H. Ney, U. Essen and R. Kneser, *On Structuring Probabilistic Dependences in Stochastic Language Modeling*, Computer, Speech, and Language, Vol. 8, pp. 1–38, 1994.

[47] T. Niesler and P. Woodland, *A Variable–Length Category–Based N–Gram Language Model*, in Proc. 1996 Int. Conf. Acoust., Speech, Sig. Proc., Atlanta, GA, pp. I164–I167, May 1996.

[48] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, *Latent Semantic Indexing: A Probabilistic Analysis*, in Proc. 17th ACM Symp. Princip. Database Syst., Seattle, WA, 1998. Also J. Comp. Syst. Sciences, 1999.

[49] F.C. Pereira, Y. Singer, and N. Tishby, *Beyond Word n-Grams*, Computational Linguistics, Vol. 22, June 1996.

[50] L.R. Rabiner, B.H. Juang, and C.-H. Lee, *An Overview of Automatic Speech Recognition*, Chapter 1 in Automatic Speech and Speaker Recognition: Advanced Topics, C.-H. Lee, F.K. Soong, and K.K. Paliwal (Eds.), Kluwer Academic Publishers, Boston, MA, pp. 1–30, 1996.

[51] R. Rosenfeld, *The CMU Statistical Language Modeling Toolkit and its Use in the 1994 ARPA CSR Evaluation*, in Proc. ARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, March 1994.

[52] R. Rosenfeld, *A Maximum Entropy Approach to Adaptive Statistical Language Modeling*, Computer Speech and Language, Vol. 10, Academic Press, London, UK, pp. 187–228, July 1996.

[53] R. Rosenfeld, *Two Decades of Statistical Language Modeling: Where Do We Go From Here*, Proc. IEEE, Spec. Issue Speech Recog. Understanding, B.H. Juang and S. Furui (Eds.), Vol. 88, No. 8, pp. 1270–1278, August 2000.

[54] R. Rosenfeld, L. Wasserman, C. Cai, and X.J. Zhu, *Interactive Feature Induction and Logistic Regression for Whole Sentence Exponential Language Models*, in Proc. Aut. Speech Recog. Understanding Workshop, Keystone, CO, pp. 231–236, December 1999.

[55] S. Roukos, *Language Representation*, Chapter 6 in Survey of the State of the Art in Human Language Technology, R. Cole (Ed.), Cambridge University Press, Cambridge, MA, 1997.

[56] R. Schwartz, T. Imai, F. Kubala, L. Nguyen, and J. Makhoul, *A Maximum Likelihood Model for Topic Classification of Broadcast News*, in Proc. Fifth Euro. Conf. Speech Comm. Technol., Rhodes, Greece, Vol. 3, pp. 1455–1458, September 1997.

[57] R.E. Story, *An Explanation of the Effectiveness of Latent Semantic Indexing by Means of a Bayesian Regression Model*, Inform. Processing & Management, Vol. 32, No. 3, pp. 329–344, 1996.

[58] J. Wu and S. Khudanpur, *Combining Nonlocal, Syntactic and N-Gram Dependencies in Language Modeling*, in Proc. Sixth Euro. Conf. Speech Comm. Technol., Budapest, Hungary, Vol. 5, pp. 2179–2182, September 1999.

[59] D.H. Younger, *Recognition and Parsing of Context–Free Languages in Time* N^3, Inform. & Control, Vol. 10, pp. 198–208, 1967.

[60] R. Zhang, E. Black, and A. Finch, *Using Detailed Linguistic Structure in Language Modeling*, in Proc. Sixth Euro. Conf. Speech Comm. Technol., Budapest, Hungary, Vol. 4, pp. 1815–1818, September 1999.

[61] X.J. Zhu, S.F. Chen, and R. Rosenfeld, *Linguistic Features for Whole Sentence Maximum Entropy Language Models*, in Proc. Sixth Euro. Conf. Speech Comm. Technol., Budapest, Hungary, Vol. 4, pp. 1807–1810, September 1999.

[62] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff, *Integration of Speech Recognition and Natural Language Processing in the MIT Voyager System*, in Proc. 1991 IEEE Int. Conf. Acoust., Speech, Signal Processing, Toronto, Canada, pp. 713–716, May 1991.

10

Semantic Information Processing of Spoken Language – How May I Help You?[sm]

A. L. Gorin, A. Abella, T. Alonso, G. Riccardi, and J. H. Wright,
AT&T Laboratories

CONTENTS

10.1 Introduction

The next generation of voice-based user interface technology enables easy-to-use automation of new and existing communication services, achieving a more natural human-machine interaction. By natural, we mean that the machine understands what people actually say, in contrast to what a system designer expects them to say. This approach is in contrast with menu-driven or strongly-prompted systems, where many users are unable or unwilling to navigate such highly structured interactions. *AT&T's 'How May I Help You?'* (HMIHY[sm]) technology shifts the burden from human to machine, wherein the system adapts to peoples' language, as contrasted with forcing users to learn the machine's jargon. The goal of such systems is to extract meaning from user's natural spoken language. It is important to quantify this notion, so that we can measure the 'semantic information content' of a spoken utterance and furthermore measure our success in extracting that information. Such a theory is crucial to being able to engineer systems that understand and act upon spoken language. The communication paradigm here involves inducing the machine to perform some action or undergo some internal transformation. A communication is deemed successful if the machine responds appropriately to the user's input. This is in contrast to the traditional goal of a communication system, which was described by Shannon [11] as follows.

> *"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another*

point. Frequently the messages have meaning, ... These semantic aspects of communication are irrelevant to the engineering problem."

In this work, the semantic aspects of communication are primary. Information theory can still be exploited, however, to provide a measure of semantic information called salience, detailed in [5] and whose utility is described later in this article. Understanding unconstrained speech is a difficult problem, both from the perspective of speech recognition and natural language understanding. Both technologies are far from perfect, especially for conversational-style speech over a telephone. The intuition underlying our approach is that for a given task, some linguistic events are crucial to recognize and understand, others not so. This intuition is quantified via salience, and we have developed algorithms [12, 3] that automatically learn the salient words, phrases and grammar fragments for a task. It can be shown empirically that these salient fragments are recognized far more reliably than average, which allows our HMIHY technology to work effectively. A tongue-in-cheek comment is that 'whoever' designed natural language did a 'good job', and made the salient events easier to recognize. A description of early laboratory experiments based on these ideas is provided in the tutorial paper [5]. In this article, we focus on two tasks involving live customer traffic in AT&T's network. In the operator services domain, the task involves placing telephone calls, specifying billing methods for those calls (e.g. collect, calling card, etc.), and requesting information about making those calls (e.g. rate, area codes, etc).

The second application is to a customer care task. In this domain, users ask questions about items on their bill, their calling-plans, account balances, etc. The customer care domain is intuitively more complex, details of which will be quantified later in this article. The primary focus of spoken language understanding (SLU) in these domains has been call-type classification, i.e. determining which service type a customer is requesting [6]. Classification is followed by routing the call to an appropriate destination, either an automated module (when available) or a human agent with some specialized skill set. We have also reported on methods for extracting named entities such as phone and credit-card numbers embedded in natural spoken language [8], and for translation into Spanish and Japanese [4]. Human/machine interactions rarely consist of a single turn. Dialog is necessary to *confirm* the machine's understanding when its confidence is low, to *clarify* ambiguities in a customer's request, and to gather additional information necessary to complete the task. For example, if someone asks to make a collect call from a train station, the ASR confidence would be low because of the noisy background, so the machine should confirm its recognition and understanding via *"Do you want to make a collect call?"* If a user asks *"Charge this call please"*, there is an ambiguity which needs to be clarified, for example via *"How do you want to charge this call, to a credit card or to a third number?"* An example of completion occurs when someone wants their account balance in customer care, whence the machine needs to know *"What is your home phone number?"*.

Traditionally, for menu-systems and strongly-prompted dialogs, the human/machine interaction is defined by a 'call-flow', essentially a long 'if-then-else' specification.

That approach does not scale well for complex natural spoken dialogs, yielding software that is difficult to design, maintain, and support. Our approach to dialog management is based on a framework called the *Construct Algebra* [2]. This theory provides the building blocks for the dialog process, comprising the relations and operations of that algebra. The result is a collection of reusable *dialog motivators*, generic rules that determine what action the dialog manager takes in its next interaction with a user, and which are portable over a range of tasks. For example, there are reusable dialog motivators for confirmation, clarification, and missing information [2].

In any domain, there is task knowledge that must be encoded and provided to the Dialog Manager and SLU modules. In HMIHY, this task knowledge is based on an object-oriented *inheritance hierarchy* [1]. This inheritance hierarchy defines the relationships amongst the call-types and named entities. For example, a customer's query about an unrecognized charge on their bill *'is a'* kind of query about a charge on their bill. That unrecognized number query *'has a'* dollar amount, item number, dialed number, etc. In computer science, it is well known how to represent these 'is a' and 'has a' relations via an object-oriented inheritance hierarchy in programming languages such as C++ or Java. The dialog manager exploits this task knowledge and the dialog motivators to govern what action to perform at each turn in the dialog.

This chapter proceeds as follows. First, we will motivate and describe the call-classification problem for automated voice services. We then describe the operator services and customer care tasks in more detail, giving examples of what people say and of what they want. Various dimensions of the linguistic and semantic complexity of these tasks will be described and quantified, including the inheritance hierarchy.We then move on to discuss language modeling for both speech recognition and spoken language understanding, including the ideas underlying automated acquisition and exploitation of salient words, phrases, and grammar fragments. The role of dialog to guide the human/machine interaction will be reviewed, including the concept of a dialog trajectory analysis. Finally, we will conclude and provide pointers to related research.

10.2 Call-Classification

In traditional telephony automation, a user is offered a list of menu options from which to select. In some cases, the desired service can be provided by simple automation, e.g. providing an account balance, or billing a call to a credit card. In other cases, the requested service can only be provided by a human agent with some specialized skill set. In either case, the menu system provides the user the opportunity to navigate to the appropriate destination where he or she can obtain service or have their problem resolved. These menus have been implemented using either touch-tone (*'press one if you want x, press two if you want y'* ...) or via voice labels

(*'please say collect, calling card'*, ...). There is also the familiar hybrid (*'press or say one if you want x'*, ...). Each of these have their place and purpose, and have proved useful when the list of options is short and well-understood by customers. When the list becomes long, however, then system designers resort to hierarchical menus, which many users are unable or unwilling to navigate. In the case of succinct menu-options that are less specific, it is often difficult for a user to decide which of the proffered categories matches what they want. There is always the trade-off in such menus of explaining each option in great detail (whence the user becomes bored and stops listening), or of being succinct in the description (whence the user cannot figure out which option to select). It is a familiar scenario for users to become frustrated and either press zero to 'bail out' of such systems, or to 'play possum' and do nothing, in the hope of being connected to a human agent.

In contrast, consider how a human receptionist would handle this same routing task. He or she would ask *'How may I help you?'*, and then the user would describe, in their own words, what is their request or problem. The receptionist's job is to know enough about the domain to transfer the caller to somebody (or something) that can provide the requested service.

Thus, we set the goal of going 'back to the future' and engineering a system with this same natural functionality. A user is greeted, and makes a request as if talking to a person. The system's job is to recognize and understand what the user wants, sufficiently to route their call to an automated module or human agent that can provide the requested service.

For example, when we collected a database of what customers say to operators, we observed that although the variation in vocabulary and language is large, most of the time they ask for one of 15 service types [6]. For example,

> *"I want to reverse the charges on this call."*
> *"Can you tell me what time it is in Tokyo?"*
> *"I was trying to call my sister and dialed a wrong number."*
> *"I've been trying to dial this number all day and can't get through."*

The first request is for a COLLECT call, the second for TIME information, and the third for a BILLING CREDIT. Automation for these three call-types is straightforward. The final query is more complex, and currently requires a person to address the problem. In any call-classification task, there is always a 'tail of the distribution' which does not fit into any of the predefined categories. We call this 'none of the above' class OTHER, and such calls are directed to a human agent.

The types of questions asked in customer care are quite different [7], where people are asking about their bills, calling-plans, etc. For example,

> *"How much money do I owe you?"*
> *"I don't recognize this phone call to Tallahassee on October 4."*
> *"What's this charge for one dollar and fifty cents?"*
> *"I have a question about my bill."*

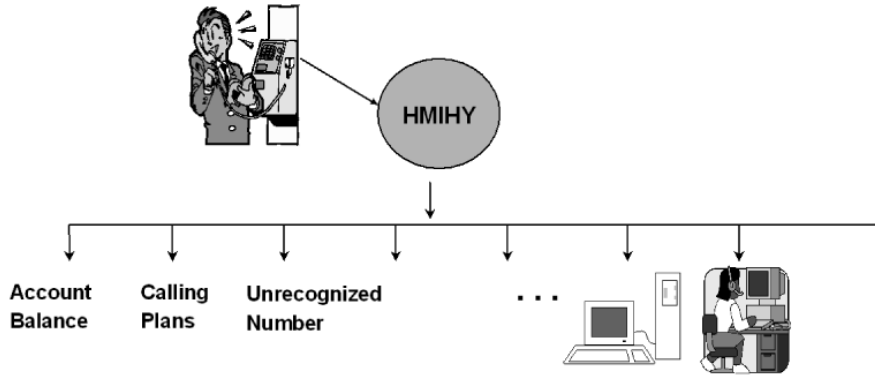

FIGURE 10.1
Call classification and routing in HMIHY.

The first query is for an ACCOUNT BALANCE and the second for an UNRECOGNIZED NUMBER, automation for both of which is straightforward and exists today. The third is somewhat vague, asking about a CHARGE ON BILL, as is the fourth, which is merely a BILLING QUERY. For these last two examples, the dialog manager must ask a clarifying question before the call can be classified and routed. Figure 10.1 illustrates the call-flow for call-routing in customer care, where the user responds to the open-ended prompts *'How may I help you?'* and is then classified and routed to an appropriate automated module or human agent that can provide the requested service.

As observed earlier, the set of semantic labels in such tasks is not a simple unstructured list. In the examples from operator services, COLLECT and CREDIT CARD are a kind of BILLING method, and any call has a FORWARD NUMBER (the number being called). Similarly, requests for RATE, TIME, or AREA CODE are all a kind of request for INFORMATION. These 'is a' and 'has a' relationships are encoded in an object oriented inheritance hierarchy, partially illustrated in Figure 10.2 for the operator services domain. The terminal nodes in the hierarchy provide sufficient specificity for the machine to 'take action', while the non-terminal nodes require clarifying queries from the dialog manager.

In comparing the operator services and customer care domains, our intuition tells us that the latter is 'more complex'. Let's explore how to quantify this intuition. A first observation is that customer utterances are significantly longer in the customer care domain. When responding to *'How may I help you?'*, the average number of words in the first domain is 19, while it is 39 words in the second. A histogram of utterance length is shown for both domains [7] in Figure 10.3. Observe that the 'shape' of the two distributions are similar - skewed unimodal with a long tail.

A second dimension of linguistic complexity is vocabulary. For a random sample of

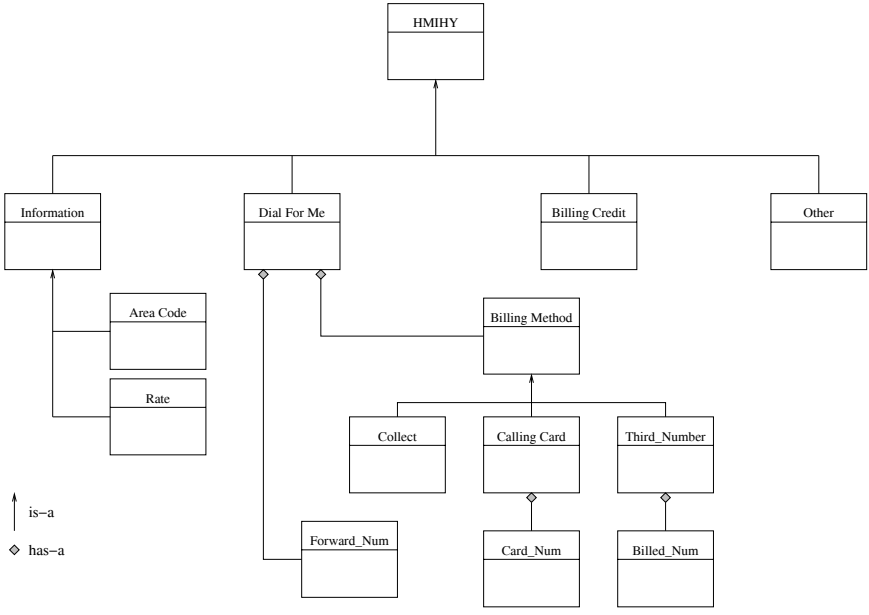

FIGURE 10.2
Inheritance hierarchy of task knowledge in operator services.

8K utterances, the accumulated vocabularies in the two domains are 3.6K and 5.2K respectively. The OOV (out of vocabulary) rate of observed new words in both cases is approximately one new word every third utterance. We observed that these OOV words are less than half proper nouns, emphasizing the high variation in customers' language for these domains. A final and traditional measure of linguistic complexity is *perplexity* [9], which can be loosely interpreted as the 'average branching factor' of the language. These are 16 and 39 respectively, again illustrating the greater complexity of the customer care domain.

One can also measure the semantic complexity of the 'classification task'. The entropy of this call-type distribution can be computed as 3.3 bits per semantic label. This provides insight into why the 'classification' problem is tractable from conversational-style speech over the telephone, while ASR remains difficult. A perplexity of 39 is equivalent to an entropy of ~ 5 bits per word, or ~ 200 bits in a 40 word utterance that we are attempting to decode via ASR. From the classification perspective, however, we are only seeking to reliably decode 3.3 bits per utterance. Although this is not a rigorous argument, it indicates why call-classification is possible, with high accuracy, while ASR is far more difficult. For example, word accuracy for these tasks is $\sim$71%; accuracy on salient phrases is much higher at $\sim$85%, and classification accuracy well exceeds 90%.

Evaluating Call Classification. Call classification can be viewed as a multi-class classification task with rejection. There are three traditional measures for such tasks

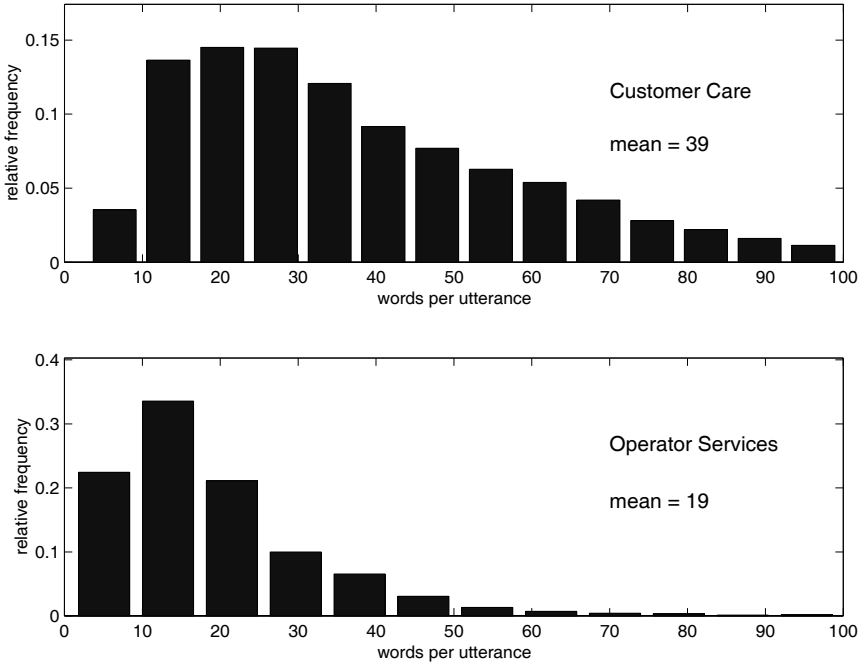

FIGURE 10.3
Histogram of utterance lengths.

[6]. First is the probability of *false rejection*, which measures how often a request for some service is rejected or classified as OTHER. A second is the probability of *correct classification*, which measures how often a classification as some call-type is correct. The third is the *true rejection rate*, which measures the probability that a request which should be classified as OTHER is indeed rejected and thus routed to a human. Dialog provides the opportunity to ask confirming and clarifying questions, thus providing improved call-classification over the case of a single utterance. While far from perfect, performance exceeding 90% has been reported in our published papers, which is far superior to customers' ability to self-select and navigate hierarchical menus. Thus, HMIHY provides both an improved user experience plus more accurate routing and thus increased automation.

Remark: There is vast literature in text categorization, for the purpose of information and document retrieval. There is also a literature on topic classification from speech, for similar purposes. Call-classification as described in this and related work has several distinguishing attributes. First, it involves speech rather than text, with the inherent difficulties of speech recognition and the disfluencies of conversational-style language. Second, the input is from cooperative users who are trying to communicate their need and make themselves understood. Third, the system has the opportunity to ask confirming or clarifying questions of that cooperative user. Fourth, there

is often collateral customer profile information available, which can be exploited in understanding a request. For example, a customer who says *"I want to know how to pay my bill"* would be routed differently if they have been delinquent in payment versus a routine request.

10.3 Language Modeling for Recognition and Understanding

For recognizing unconstrained spoken language, the state-of-the-art involves training a stochastic language model which predicts the probability of a sequence of words. For example, given a sentence $S = v_1 v_2 \ldots v_n$, we want to estimate the probability of the word v_i given the history of preceding words, i.e. $P(v_i | v_1 v_2 \ldots v_{i-1})$. It is not tractable to estimate these probabilities for all possible histories due to data sparseness. So, the most familiar method is the n-gram model, which estimates the probability of a word given only the preceding n-1 words (where typically n=3, denoted a trigram language model). As n increases, the memory and computation requirements of the language model increase, as does data sparseness. An alternative is to selectively introduce longer-range 'history' in the form of variable length units [9]. For ASR language modeling, these units are selected on the basis of entropy minimization, leading to 'superwords' such as *'I want to make a'* , *'collect call'* and *'card call'* in the operator services domain. In that case, a bigram language model on these variable length units would lead to estimates such as $P(\text{'collect call'} \mid \text{'I want to make a'})$, effectively providing a 7-gram model, but only selectively. In [9], it was shown how to embed these acquired phrases into a stochastic finite state automaton to provide language models for ASR. It was shown that these models had a similar accuracy to high-order n-gram models, but with the computation and memory requirements of low-order models.

After recognizing the words spoken by a user, the next step is to 'understand' what they said. Our early experiments focused on methods based on a 'bag of words' model [5]. We discovered, however, that ignoring the temporal order of the words in an utterance was not optimal. Language does indeed have structure which can be exploited to enable more reliable understanding.

The first stage was to investigate and develop algorithms to automatically acquire salient phrases for a task, i.e., while *'wrong'* is a salient word in the operator services task (associated to needing a billing credit), *'wrong number'* is even more salient, and *'dialed a wrong number'* more salient still [6]. Salient phrases are preferable to words because they have sharper semantics, and because longer events are more reliably recognized in speech.

The second stage commenced with observing that many of these salient phrases were similar, such as *'dialed a wrong number'* and *'dialed the wrong number'*, leading to the development of clustering algorithms exploiting a combination of string-edit distances and semantic distortions [12]. These clusters of salient phrases are compactly

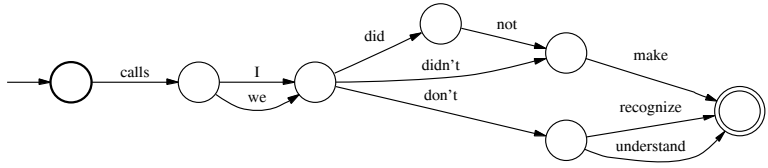

FIGURE 10.4
A salient grammar fragment.

represented as finite state machines, and denoted *salient grammar fragments*. Clusters of phrases are preferable because of parsimony, enabling pooling of statistics across multiple low frequency phrases. These grammar fragments are also advantageous because they are robust to ASR errors 'within the fragment'.

In Figure 10.4, a salient grammar fragment from the customer care task is shown, which is strongly associated with queries about an UNRECOGNIZED NUMBER on a bill.

These methods were extended to include hierarchical clustering by also exploiting syntactic distortions [3]. Finally, we observe that embedding salient phrases in the ASR language model has been shown to improve classification performance, while having negligible effect on word accuracy.

To classify an utterance, these salient grammar fragments are matched against the ASR output, then a decision rule applied to combine the lattice of detections and their associations. This is illustrated below, showing the transcription of a customer utterance, then the ASR output with detected salient fragments highlighted. There are three detected fragments: one associated with the call-type of CELLULAR , the other two associated with CALLING PLAN. Based on strength of associations plus coverage, the output of the SLU is thus the latter.

- **User**: okay I got AT&T Wireless phones and when I got them he told me that I would be switched to 7 cents a minute for all my AT&T long distance calling because I was on 10 10 cents One Rate plan

- ***ASR***: *yeah I'm not AT&T WIRELESS_PHONE and when I got and she told me that I would be switched to 7_CENTS_A_MINUTES_FOR_ALL my AT&T long distance on that I was on 10 10 cents ONE_RATE_PLAN*

- **SLU**: Calling_Plans

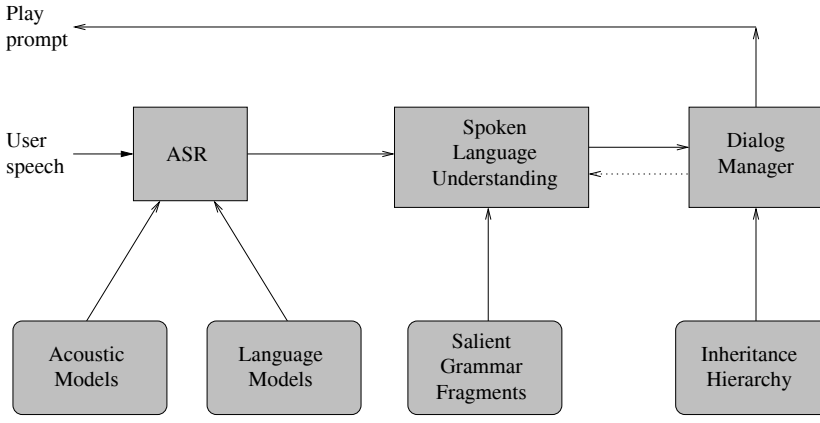

FIGURE 10.5
Natural spoken dialog in HMIHY.

10.4 Dialog

An interaction with HMIHY often comprises more than one turn, with dialog necessary to confirm low-confidence interpretations, to clarify ambiguities, and to request missing information to complete a transaction. Figure 10.5 illustrates the architecture of the HMIHY dialog system, including an ASR component which requires acoustic and language models, an SLU component which requires the salient grammar fragments, and the dialog manager which acts upon the SLU-output plus the task knowledge in the inheritance hierarchy.

The ASR language model must vary over the dialog states, since what users say of course changes with respect to context. We refer the reader to [10] for a detailed analysis of such variations in spoken language over the course of a dialog.

We provide two dialog examples, the first illustrating two invocations of the 'missing information' motivator to obtain the billing method and card number, the second illustrating the 'clarification' motivator.

Machine: AT&T. How may I help you?
User: I need a credit, I got the wrong number.
Machine: What was the number that you dialed?
User: I dialed four one five two eight .....
Machine: Was the call billed to the phone that you're calling from now?
User: No, to a credit card.
Machine: What was the card number?
User: four five six seven .....
Machine: I've given you credit for that call. Thank you for using AT&T.

Machine: AT&T. How may I help you?
User: I have a question about my June bill.
Machine: Okay, what is your question?
User: There's a call to Tallahassee, but I don't know anyone there.
Machine: I can help you with that.
Are you calling from your home phone?
User: I'm calling from my house.
Machine: Please hold on while I transfer your call.

10.5 Conclusions

In summary, we have described AT&T's HMIHY technology which embodies the next generation of voice-enabled services. These advances in speech, language, and dialog technology shift the burden from user to machine, where the machine adapts to customers' spoken language, in contrast to forcing people to learn the machine's jargon. In developing such a 'first of its kind' system, many research issues have arisen. The interested reader can access our web site `www.research.att.com/~algor/hmihy`, which contains a link to an on-line collection of many of our research papers.

References

[1] A. Abella and A.L. Gorin, "Generating Semantically Consistant Inputs to a Dialog Manager," Proc. Eurospeech, Greece, pp. 1879-1882, Sept. 1997.

[2] A. Abella and A. Gorin, "Construct Algebra: Analytical Dialog Management," Proc. ACL, Washington D.C., June 1999.

[3] K. Arai, J.H. Wright, G. Riccardi and A. Gorin, "Grammar Fragment Acquisition Using Syntactic and Semantic Clustering," Speech Communication, vol. 27(1), Jan. 1999.

[4] S. Bangalore and G. Riccardi, "Stochastic Finite-State Models for Spoken Language Machine Translation," Proc. NAACL, Seattle, WA, May 2000.

[5] A.L. Gorin, "On Automated Language Acquisition," 97(6), pp. 3441-3461, Journal of the Acoustical Society of America, June 1995.

[6] A.L. Gorin, G. Riccardi and J.H. Wright, "How May I Help You?," Speech Communication, vol. 23, pp. 113-127, 1997.

[7] A.L. Gorin, J. H. Wright, G. Riccardi, A. Abella, and T. Alonso, "Semantic Information Processing of Spoken Language," Proc. ATR Workshop on Multilingual Speech Communication, Kyoto, Japan, Oct. 2000.

[8] M. Rahim, G. Riccardi, L. Saul, J. Wright, B. Buntschuh, and A. Gorin, "Robust Numeric Recognition in Spoken Language Dialog," Speech Communication, vol. 34 (1-2), pp. 195-212, 2001.

[9] G. Riccardi, R. Pieraccini, and E. Bocchieri, "Stochastic Automata for Language Modeling," Computer Speech and Language, 10(4), pp. 265-293, 1996.

[10] G. Riccardi and A.L. Gorin, "Spoken Language Adaptation over Time and State in a Natural Spoken Dialog System," IEEE Trans. on Speech and Audio, vol. 8, pp. 3-10, Jan. 2000.

[11] C. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, Vol. XXVII, No. 3, pp. 379-423, July 1948.

[12] J.H. Wright, A.L. Gorin and G. Riccardi, "Automatic Acquisition of Salient Grammar Fragments for Call-Type Classification," Proc. Eurospeech, Greece, 1997.

11

Machine Translation Using Statistical Modeling

Herman Ney, and F. J. Och
Aachen University of Technology, Germany

CONTENTS

Abstract. This chapter gives an overview of the statistical approach to machine translation, in particular the translation of spoken dialogues in the framework of the VERBMOBIL project. Starting with the Bayes decision rule as in speech recognition, we show how the required probability distributions can be structured into three parts: the language model, the alignment model, and the lexicon model. We describe the components of the system and report results on the VERBMOBIL task. The experience obtained in the VERBMOBIL project, in particular in the final evaluation, showed that the statistical approach resulted in significantly lower error rates than three competing translation approaches: the sentence error rate was 29% in comparison with 52% to 62% for the other translation approaches. Finally, we discuss the integrated approach to speech translation as opposed to the serial approach as it is widely used nowadays.

11.1 Introduction

The automatic translation of language is generally referred to as *machine translation*. Typically, this term is used for *written language* or *text* input, where the implicit assumption is that the input is uncorrupted, i.e. without errors. This task is very much different from *spoken speech* input, where the system must cope with speech

recognition errors and also the ungrammatical structure of spoken language.
The translation of *spontaneous speech* poses additional difficulties for the task of automatic translation. Typically, these difficulties are caused by errors of the recognition process, which is carried out before the translation process. As a result, the sentence to be translated is not necessarily well-formed from a syntactic point-of-view. Even without recognition errors, speech translation has to cope with a lack of conventional syntactic structures because the structures of spontaneous speech differ from that of written language.
The statistical approach shows the potential to tackle these problems for the following reasons. First, the statistical approach is able to avoid hard decisions at any level of the translation process. Second, for any source sentence, a translated sentence in the target language is guaranteed to be generated. In most cases, this will be hopefully a syntactically perfect sentence in the target language; but even if this is not the case, in most cases, the translated sentence will convey the meaning of the spoken sentence.
The organization of this chapter is as follows:

- Section 2: **Statistical Decision Theory and Linguistics.**
  We will present the Bayes decision rule and the resulting architecture for the translation of written language.

- Section 3: **Alignment and Lexicon Models.**
  A key component in the statistical approach is the so-called alignment concept, which is similar to hidden Markov models used in speech recognition and which will be considered in more detail.

- Section 4: **Alignment Templates: From Single Word to Word Groups.**
  To introduce more context into the translation process, we will consider the method of alignment templates that allows us to translate word groups or phrases as a whole.

- Section 5: **Experimental Results**.
  Although the methods presented apply both to *written* and *spoken* language, we will limit ourselves here to spoken language and report on the final experimental evaluation that were carried out in the VERBMOBIL project.

- Section 6: **Speech Translation: The Integrated Approach.**
  As an alternative to the *serial* coupling of recognition and translation that is used in our and other systems as well, we will consider the *integrated* approach to recognition and translation and the corresponding form of the Bayes decision rule [26].

Whereas statistical modelling is widely used in speech recognition and it is impossible to enumerate all systems, there seem to be only a small number of research groups that have applied statistical modelling to the translation of written or spoken language [1, 7, 21, 50]. The presentation here is based on work carried out in the framework of the EUTRANS project [19] and the VERBMOBIL project [49].

11.2 Statistical Decision Theory and Linguistics

11.2.1 The Statistical Approach

The use of statistics in computational linguistics has been extremely controversial for more than three decades. The controversy is very well summarized by the statement of Chomsky in 1969 [15]:

> "... it must be recognized that the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term".

This statement was considered to be true by the majority of experts from artificial intelligence and computational linguistics, and the concept of statistics was banned from computational linguistics for many years.
What is overlooked in this statement is the fact that, in an automatic system for speech recognition or text translation, we are faced with the problem of making decisions. It is exactly here where statistical decision theory comes in. In speech recognition, the success of the statistical approach is based on the equation:

$$\begin{aligned}\text{Speech Recognition} &= \text{Acoustic–Linguistic Modelling} \\ &\quad + \text{Statistical Decision Theory}\end{aligned}$$

Similarly, for machine translation, the statistical approach is expressed by the equation:

$$\begin{aligned}\text{Machine Translation} &= \text{Linguistic Modelling} \\ &\quad + \text{Statistical Decision Theory}\end{aligned}$$

For the 'low-level' description of speech and image signals, it is widely accepted that the statistical framework allows an efficient coupling between the observations and the models, which is often described by the buzz word 'subsymbolic processing'. But there is another advantage in using probability distributions in that they offer an explicit formalism for expressing and combining hypothesis scores:

- The probabilities are directly used as scores. These scores are normalized, which is a desirable property: when increasing the score for a certain element in the set of all hypotheses, there must be one or several other elements whose scores are reduced at the same time.
- It is straightforward to combine scores. Depending on the task, the probabilities are either multiplied or added.
- Weak and vague dependences can be modelled easily. Especially in spoken and written natural language, there are nuances and shades that require 'grey levels' between 0 and 1.

11.2.2 Bayes Decision Rule for Written Language Translation

In machine translation for written language, the goal is the translation of a text given in a source language into a target language. We are given a source string $f_1^J = f_1 ... f_j ... f_J$, which is to be translated into a target string $e_1^I = e_1 ... e_i ... e_I$. For historical reasons [12], we use the symbols f (like French) for source words and the symbol e (like English) for target words. In this chapter, the term *word* always refers to a *full-form* word. Among all possible target strings, we will choose the string with the highest probability which is given by Bayes decision rule [12]:

$$\begin{aligned} \hat{e}_1^I &= \arg\max_{e_1^I} \{Pr(e_1^I | f_1^J)\} \\ &= \arg\max_{e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J | e_1^I)\} \end{aligned}$$

Here, $Pr(e_1^I)$ is the language model of the target language, and $Pr(f_1^J | e_1^I)$ is the string translation model which will be decomposed into lexicon and alignment models. The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. The overall architecture of the statistical translation approach is summarized in Figure 11.1.
In general, as shown in this figure, there may be additional transformations to make the translation task simpler for the algorithm. The transformations may range from the categorization of single words and word groups to more complex preprocessing steps that require some parsing of the source string. We have to keep in mind that in the search procedure both the language and the translation model are applied *after* the text transformation steps. However, to keep the notation simple, we will not make this explicit distinction in the subsequent exposition.

11.2.3 Related Approaches

There are a number of related approaches that are also corpus-based and therefore closely related to the statistical approach:

- finite-state approaches [2, 4, 14, 46]:
  Here, the probabilistic dependences are represented by finite-state structures that can be learned automatically from training data.

- example-based approaches [3, 39, 42]:
  In example-based approaches, large bilingual chunks are excised from the set of bilingual sentence pairs. In the translation process, the most similar chunk in the set of source-language chunks is determined, and its corresponding target-language chunk is used as translation. This baseline variant may be refined in various ways to introduce generalization capabilities (see also Section 11.5.4).

- syntax-based statistical approaches [1, 52, 53]:
  These approaches are obtained as an extension of the statistical approach,

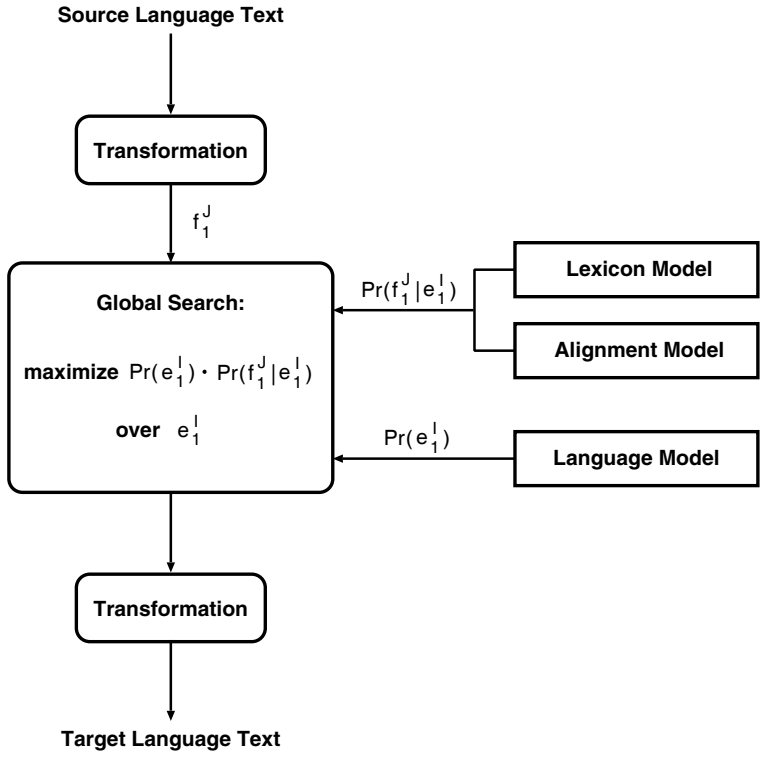

FIGURE 11.1
Architecture of the translation approach based on Bayes decision rule.

where syntactic structures are incorporated into the baseline statistical approach, in particular the so-called alignment models (see later). The syntactic structure may be modelled in the target language only or in both target and source language.

11.3 Alignment and Lexicon Models

11.3.1 Concept of Alignment Modelling

A key issue in modelling the string translation probability $Pr(f_1^J|e_1^I)$ is the question of how we define the correspondence between the words of the target sentence and the words of the source sentence. In typical cases, we can assume a sort of pairwise dependence by considering all word pairs (f_j, e_i) for a given sentence pair $(f_1^J; e_1^I)$.

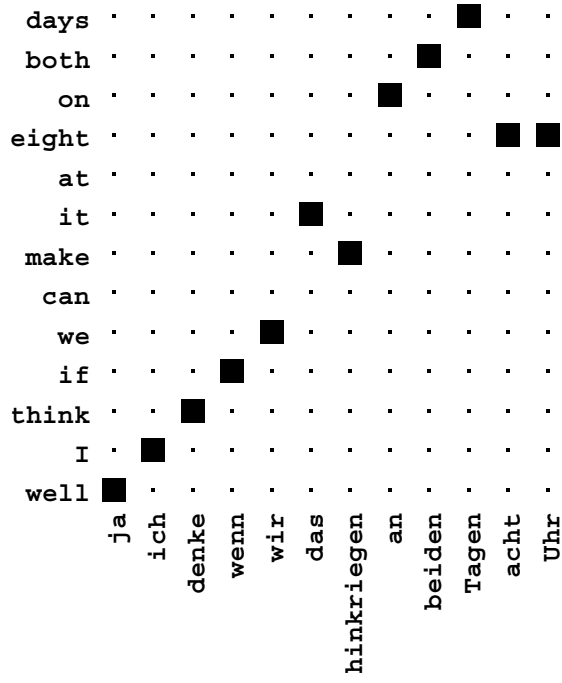

FIGURE 11.2
Example of an alignment for a German-English sentence pair.

Here, we will further constrain this model by assigning each source word to *exactly one* target word. Models describing these types of dependences are referred to as *alignment models* [12, 16, 48].

When aligning the words in parallel texts, we typically observe a strong localization effect. Figure 11.2 illustrates this effect a German–English sentence pair from the VERBMOBIL corpus. In many cases, although not always, there is an additional property: over large portions of the source string, the alignment is monotone. In the following, we will consider two approaches to alignment modelling in more detail, namely hidden Markov models and models IBM 1–5.

11.3.2 Hidden Markov Models

The first approach to alignment modelling will be based on hidden Markov models (HMM) as they have been used successfully in speech recognition for a long time [22, chapter 2], [37, chapter 6]. Thus the aligment mapping in translation is similar to the time alignment path (or state sequence) in speech recognition.

To arrive at a quantitative specification, we first define the

$$\text{alignment mapping:} \quad j \rightarrow i = a_j$$

which assigns a word f_j in position j to a word e_i in position $i = a_j$. The general concept of word alignments was introduced in [12].
Using the same basic principles as in HMMs for speech recognition, we can rewrite the probability by introducing the 'hidden' alignments $a_1^J := a_1 ... a_j ... a_J$ for each sentence pair $[f_1^J; e_1^I]$:

$$Pr(f_1^J|e_1^I) = \sum_{a_1^J} Pr(f_1^J, a_1^J|e_1^I)$$

To clarify the meaning of the term *'hidden'* in comparison with speech recognition, we note that the model states as such (representing words) are *not* hidden but the actual alignments, i.e. the *sequence* of position index pairs (j, i) with $i = a_j$. To draw the analogy with speech recognition, we have to identify the states (along the vertical axis) with the positions i of the target words e_i and the time (along the horizontal axis) with the positions j of the source words f_j.
We can decompose the probability distribution $Pr(f_1^J, a_1^J|e_1^I)$ as follows:

$$\begin{aligned}
& Pr(f_1^J, a_1^J|e_1^I) \; = \\
& = Pr(J|e_1^I) \cdot Pr(f_1^J, a_1^J|e_1^I, J) \\
& = Pr(J|e_1^I) \cdot \prod_{j=1}^{J} \Big[Pr(f_j, a_j|a_1^{j-1}, f_1^{j-1}, e_1^I, J) \Big] \\
& = Pr(J|e_1^I) \cdot \prod_{j=1}^{J} \Big[Pr(a_j|a_1^{j-1}, f_1^{j-1}, e_1^I, J) \cdot Pr(f_j|f_1^{j-1}, a_1^j, e_1^I, J) \Big]
\end{aligned}$$

The above formulation does not make any assumptions about the dependences in the probability distribution and results in three distributions which need further specifications: the length model $Pr(J|e_1^I)$, the alignment model $Pr(a_j|a_1^{j-1}, f_1^{j-1}, e_1^I, J)$, and the lexicon model $Pr(f_j|f_1^{j-1}, a_1^j, e_1^I, J)$. These models are too general to be used directly and, in the following, we will limit the dependences in these models, e.g., we will assume first-order over even zero-order dependences for the conditional probability $Pr(a_j|a_1^{j-1}, f_1^{j-1}, e_1^I, J)$. We consider three cases:

- **baseline HMM**:
  By looking at real alignments for sentence pairs, it is evident that the mathematical model should try to capture the strong dependence of a_j on the preceding alignment. Therefore, when simplifying the dependences in the alignment model, we would like to retain the dependence of $Pr(a_j|\cdot)$ on the position a_{j-1} of the immediate predecessor. Thus we obtain the alignment model:

$$Pr(a_j|a_1^{j-1}, f_1^{j-1}, e_1^I, J) := p(a_j|a_{j-1}, I, J)$$

where we also have retained the dependence on the length J of the observed source sentence and the length I of the hypothesized target sentence.

For the lexicon model $Pr(f_j|\cdot)$, we make the assumption that the dependence is limited to the target word e_i with $i = a_j$, i.e. e_{a_j}, and nothing else:

$$Pr(f_j|f_1^{j-1}, a_1^j, e_1^I, J) := p(f_j|e_{a_j})$$

Finally, for the length model $Pr(J|\cdot)$, we assume a dependence on the length J of the source sentence f_1^J only:

$$Pr(J|e_1^I) := p(J|I)$$

We mention that the length model has been included for the sake of completeness and is not very important in practice. In speech recognition, there is typically no length model. Instead, a special symbol for sentence end is added to the vocabulary.

- **homogeneous HMM**
  To render the alignment probability independent of absolute positions and also to reduce the number of alignment parameters [16, 48], we assume that the alignment probabilities $p(a_j|a_{j-1}, I, J)$ depend only on the jump width $(a_j - a_{j-1})$ and nothing else. Using $i = a_j$ and $i' = a_{j-1}$, we have:

$$\begin{aligned} Pr(a_j|a_1^{j-1}, f_1^{j-1}, e_1^I, J) &= p(a_j|a_{j-1}, I, J) \\ &:= \frac{q(a_j - a_{j-1})}{\sum_{i=1}^{I} q(i - a_{j-1})} \end{aligned}$$

  with a non-negative table $q(.)$:

$$\Delta i \equiv i - i' \quad \rightarrow \quad q(\Delta i)$$

  which has to be estimated from the bilingual training corpus (like the free parameters of the other distributions introduced).

- **context dependent HMM**:
  It can be argued that, for good models, more context should be captured in the dependences.

  Thus, we extend the alignment model:

$$Pr(a_j|a_1^{j-1}, f_1^{j-1}, e_1^I, J) := p(a_j|a_{j-1}, I, J; f_{j-1})$$

  So, in comparison with the baseline model, there is an additional dependence on the source word f_{j-1} in position $(j - 1)$. First experiments with such a type of model are reported in [32].

The lexicon model can be extended in a similar way:

$$Pr(f_j|f_1^{j-1}, a_1^j, e_1^I, J) := p(f_j|e_{a_j}; f_{j-1}, e_{a_{j-1}})$$

Here, the dependences have been extended to f_{j-1} and $e_{a_{j-1}}$. Instead of these additional words themselves, parts-of-speech classes or automatically trained word classes could be used (see later) [13, 23, 31]. Such a type of extended lexicon model does not seem to have been tested yet experimentally.

11.3.3 Models IBM 1–5

The historical development of statistical machine translation was slightly different from this presentation in that the models IBM 1–5 were introduced for alignment modelling *before* HMMs were used. The models IBM 1–5 were introduced in [12] as a series of alignment models with increasing complexity.

- **models IBM-1 and IBM-2: zero-oder dependence.**
  Rather than a *first-order* dependence, we can also use a *zero-order* model for the alignment model, where there is only a dependence on the *absolute* position index j of the source string:

  $$Pr(a_j|a_1^{j-1}, f_1^{j-1}, e_1^I, J) := p(a_j|j, I, J)$$

  Both the length model and the lexicon model are the same as for the HMM. For such a zero-order model, it can be shown [12] that we have the following identity:

  $$\begin{aligned} Pr(f_1^J|e_1^I) &= p(J|I) \cdot \sum_{a_1^J} \prod_{j=1}^{J} [p(a_j|j, I, J) \cdot p(f_j|e_{a_j})] \\ &= \ldots \\ &= p(J|I) \cdot \prod_{j=1}^{J} \sum_{i=1}^{I} [p(i|j, I, J) \cdot p(f_j|e_i)] \end{aligned}$$

  The sum in the last equation can be interpreted as a mixture-type distribution with mixture weights $p(i|j, I, J)$ as alignment probabilities and with component distributions $p(f_j|e_i)$ as lexicon probabilities. The model IBM-1 is a special case with a uniform alignment probability:

  $$p(i|j, I, J) = \frac{1}{I}$$

  The presentation so far has not used the so-called 'empty word' [12]. The empty word is added to the target sentence e_1^I to allow for source words which have no direct counterpart in the target sentence e_1^I. Formally, the concept of the empty word is incorporated into the alignment models by adding the empty word e_0 at position $i = 0$ to the target sentence e_1^I and aligning all source words f_j without a direct translation to this empty word.

- **model IBM-3: fertility concept.**
  As introduced in [12], the alignment model can be extended by the concept of fertility. The idea is that often a word in the target language may be aligned to several words in the source language. This extension results in the so-called model IBM-3. For each target word e, there is a probability distribution over its possible fertilities ϕ:

  $$p(\phi|e)$$

  Experimentally, we observe that the fertilities take on values from 0 to 4. For a given alignment a_1^J, we compute the fertility ϕ_i of a target word e_i in position i as the number of aligned source words f_j with $a_j = i$:

  $$\text{fertility:} \quad \phi_i := \sum_j \delta(a_j, i)$$

  Using this equation, we can start with an HMM or model IBM-2 and then compute initial values for the fertilities. In particular, the fertility concept can be used to better model target words having no counterpart in the source sentence, i.e. target words e_i with fertility $\phi_i = 0$.

- **models IBM-4 and IBM-5: inverted alignments with first-order dependence.**
  For space limitations, we can give only a simplified description of these models. To obtain these models, we assume that the probability distribution $Pr(f_1^J, a_1^J|e_1^I)$ is the result of a process consisting of three steps, each of which involves a (simple) probability distribution. The first step is the selection of a fertility ϕ_i for each (hypothesized) target word $e_i, i = 1, ..., I$. In the next step, for each target word e_i, we generate the set of associated source words f according to the fertility ϕ_i, where the (final) positions are not specified yet. In the third step, the source words are permuted so that the observed sequence f_1^J is produced. The main advantage of the above interpretation is that, as we will see later, it is better suited for a search strategy that builds up partial string hypotheses e_1^i over target positions $i = 1, ..., I$.

  As a result, we have a sort of inverted alignments, i.e. a mapping from the target positions i to the source positions j:

  $$\text{inverted alignment mapping:} \quad i \rightarrow j = b_i$$

  which in [12] is referred to as distortion model. For these inverted alignments $b_1^I := b_1, ..., b_i, ..., b_I$, we assume a first-order dependence as for the HMM:

  $$p\Big(b_i|b_{i-1}, f_{b_i}, e_{i-1}\Big)$$

  Here, there is an additional dependence on the word context that is captured by the source word f_j in position $j = b_i$ and the target word e_{i-1}. To really apply

the above probability model, several refinements are needed. First, we must take into account that the fertility of word e_i in position i may be different from 1, e.g., for a fertility larger than 1, several positions on the target axis j have to be produced. Second, the dependence on b_{i-1} does not use the *absolute* positions, but only *relative* positions. Thus, we have a dependence on the 'jump width' $\Delta j = b_i - b_{i-1}$ along the source axis j as for the homogeneous HMM along the target axis i. Third, to reduce the number of free parameters, the dependence on the words f_{b_i} and e_{i-1} is replaced by a dependence on the corresponding parts-of-speech or word classes [13, 23, 31]: $G(f_{b_i})$ and $G(e_{i-1})$:

$$p\Big(b_i|b_{i-1}, G(f_{b_i}), G(e_{i-1})\Big)$$

These word classes can be trained separately for target and source language [13, 23]. or jointly for both languages [31]. The resulting approach is referred to as model IBM-4.

Remarkably enough, the model IBM-4 is not normalized as each probability distribution should be because it puts probability mass on events that can never occur (for more details see [12]). From the model IBM-4, we obtain the model IBM-5 by enforcing the strict normalization of the probabilities. The resulting model can be summarized as follows. We imagine that the source positions are covered in a left-to-right strategy, where occasionally some of the source positions can be skipped. To keep track of the occupied source positions, the probability of the inverted alignment is made dependent on the whole history for the partial alignment b_1^{i-1}. For a vacant position $j = b_i$, we have:

$$p\left(b_i|b_1^{i-1}, G(f_{b_i})\right)$$

To be more exact, we note that the dependence on b_1^{i-1} is mainly limited to the number of free source positions $V(b_1^{i-1})$ and to the number of free source positions between b_{i-1} and b_i. In comparison with model IBM-4, the dependence on the preceding target word e_{i-1} has been dropped to reduce the number of free parameters.

Although some of the above models take one-to-many alignments explicitly into account, the lexicon probabilities $p(f|e)$ are still based on single words in each of the two languages. The lexicon model presented so far is very simple. In reality, the translation of a word may depend on the details of the *word context*. To capture these types of dependences, maximum entropy models were proposed [9, 20].

11.3.4 Training

The free parameters of the probability distributions introduced are estimated from a corpus of bilingual sentence pairs. The training criterion is the maximum likelihood criterion. Since the models that have been introduced are complex, the training algorithms can guarantee only local convergence. In order to mitigate the problems with

poor local optima, we apply the concept presented in [12]. The training procedure is started with a simple model for which the problem of local optima does not occur or is not critical. In particular, the model IBM-1 has the advantage that it has only a single optimum and thus convergence problems cannot exist [12]. The parameters of the simple model are then used to initialize the training procedure of a more complex model. In such a way, a series of models with increasing complexity can be trained. Typical sequences are [IBM-1,-2,-4,-5] or [IBM-1, HMM].
The training procedure is based on the maximum likelihood criterion, which however can be used only in an iterative way. For the models IBM-1, IBM-2 and HMM, this is the so-called expectation-maximization algorithm for which a closed-form solution is available within each iteration. For the other models, namely IBM-3, IBM-4, and IBM-5, this is not the case anymore, and even within each iteration, numerical approximations have to be used [12, 33]. What has been said so far goes for the *exact* likelihood criterion, where we sum over *all* possible alignments. When instead we use the *maximum approximation*, where only the best alignment is considered, the situation might be very much different in that some of the problems go away. However, there have not yet been many systematic studies on how much we lose by the maximum approximation [33].
In systematic experiments, it was found that the quality of the alignments determined from the bilingual training corpus has a direct effect on the translation quality [32]. By exchanging the role of target and source language in the training procedure, we found that the quality of the alignments could be significantly improved.
From a general point of view, the alignments can be interpreted as a method for finding words or word groups that are equivalent in source language and target language. After these equivalences have been found, they may be modelled in various, data-driven approaches to build a translation system. In this chapter, we will consider the so-called alignment templates (see later), but these equivalences may as well be used in finite-state transducers [14].

11.3.5 Search

The task of the search algorithm is to generate the most likely target sentence e_1^I of unknown length I for an observed source sentence f_1^J. The search must make use of all three knowledge sources as illustrated by Figure 11.3: the alignment model, the (bilingual) lexicon model and the language model. All three of them must contribute in the final decision about the words in the target language.
To illustrate the specific details of the search problem, we specify the alignment model in more detail:

- we use *inverted* alignments as in the model IBM-4 [12] which define a mapping from *target* to *source* positions rather than the other way round.
- we allow *several* positions in the source language to be covered, i.e. we consider mappings B of the form:

$$B : i \rightarrow B_i \subset \{1, ..., j, ..., J\}$$

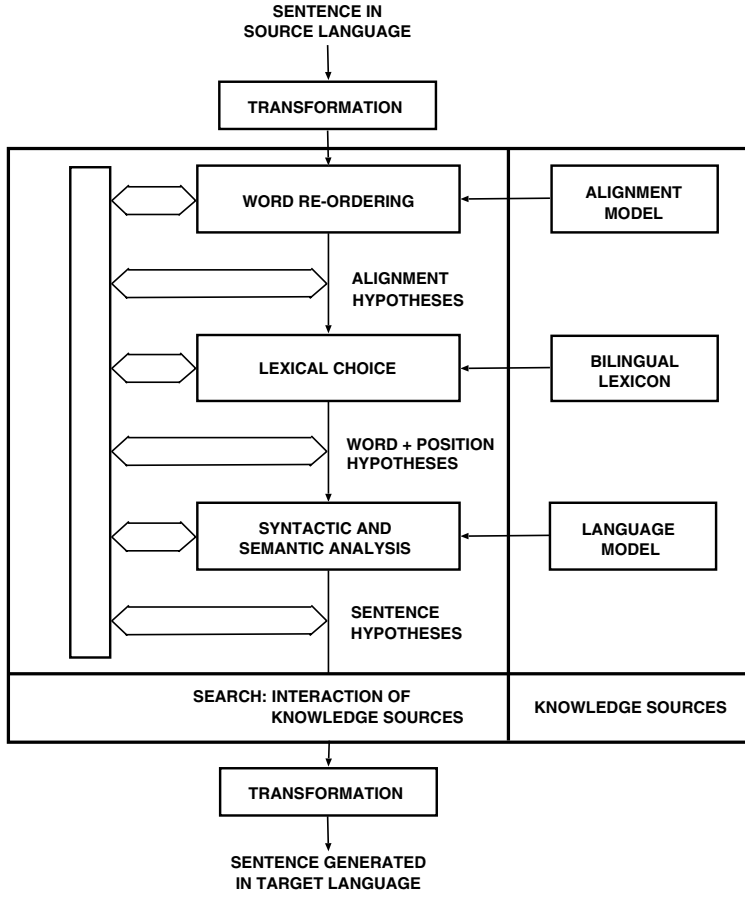

FIGURE 11.3
Illustration of search in statistical translation.

For this inverted alignment mapping with *sets* B_i of source positions, we again assume a sort of first-order model:

$$p(B_i|B_{i-1}, e_{i-1})$$

where we have dropped the dependence on I and J.
We replace the sum over all alignments by the best alignment, which is referred to as maximum approximation in speech recognition. Using a trigram language model $p(e_i|e_{i-2}, e_{i-1})$, we obtain the following search criterion:

$$\max_I \left\{ p(J|I) \cdot \max_{B_1^I, e_1^I} \prod_{i=1}^{I} \left(p(e_i|e_{i-2}^{i-1}) \cdot p(B_i|B_{i-1}, e_{i-1}) \cdot \prod_{j \in B_i} p(f_j|e_i) \right) \right\}$$

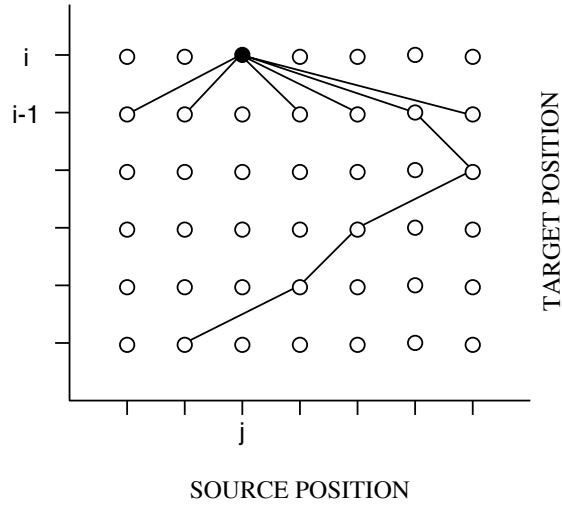

FIGURE 11.4
Illustration of bottom-to-top search.

Considering this criterion, we can see that we can build up hypotheses of partial target sentences in a *bottom-to-top* strategy over the positions i of the target sentence e_1^i as illustrated in Figure 11.4. An important constraint for the alignment is that *all* positions of the source sentence should be covered exactly *once*. This constraint is similar to that of the traveling salesman problem where each city has to be visited exactly once. Details on various search strategies can be found in [7, 27, 30, 35].

The type of language model we use ranges from a trigram to a 5-gram, which can be either word- or class-based. Beam search is used to handle the huge search space. To normalize the costs of partial hypotheses covering different parts of the input sentence, an (optimistic) estimation of the remaining cost is added to the current accumulated cost as follows. For each word in the source sentence, a lower bound on its translation cost is determined beforehand. Using this lower bound, it is possible to achieve an efficient estimation of the remaining cost [35, 44]. For other papers on the search process in translation, the reader is referred to [8, 21, 24, 50].

11.3.6 Algorithmic Differences between Speech Recognition and Language Translation

It is interesting to consider the differences between the algorithms for speech recognition and those for machine translation:

- monotonicity:
  In speech recognition, there is a strict monotonicity between the sequence of acoustic vectors and the sequence of recognized words or phonemes. This is not the case for machine translation, and therefore the search problem becomes more complicated.

- fertility:
  In machine translation, we have to decide whether a word is present in the target string or not. Therefore, it is important to assign a fertility to each word of the target vocabulary. In speech recognition, the counterpart of a word is an HMM state. However, we never take decisions about states, but about whole phoneme models either with or without context. Therefore the concept of fertility is not really needed in speech recognition.

11.4 Alignment Templates: From Single Words to Word Groups

11.4.1 Concept

A general shortcoming of the baseline alignment models is that they are mainly designed to model the lexicon dependences between single words. Therefore, we extend the approach to handle word groups or phrases rather than single words as the basis for the alignment models [34]. In other words, a whole group of adjacent words in the source sentence may be aligned with a whole group of adjacent words in the target language. As a result, the context of words tends to be explicitly taken into account, and the differences in local word orders between source and target languages can be learned explicitly. Figure 11.5 shows some of the extracted alignment templates for a sentence pair from the VERBMOBIL training corpus. The training algorithm for the alignment templates extracts all phrase pairs which are aligned in the training corpus up to a maximum length of seven words. To improve the generalization capability of the alignment templates, the templates are determined for bilingual word classes rather than words directly. These word classes are determined by an automatic clustering procedure [31].
A general deficiency of the baseline alignment models is that they are able to model correspondences only between single words. A first countermeasure was the refined alignment model used in the quasi-monotone search. A more systematic approach is to consider word groups rather than single words as the basis for the alignment models. In other words, a whole group of adjacent words in the source sentence may be aligned with a whole group of adjacent words in the target language [34]. Such a mapping will be referred to as *alignment template* in the following. Example of such alignment templates are shown in Figure 11.5; these examples were real experimental results obtained by the method to be presented. As can be seen from

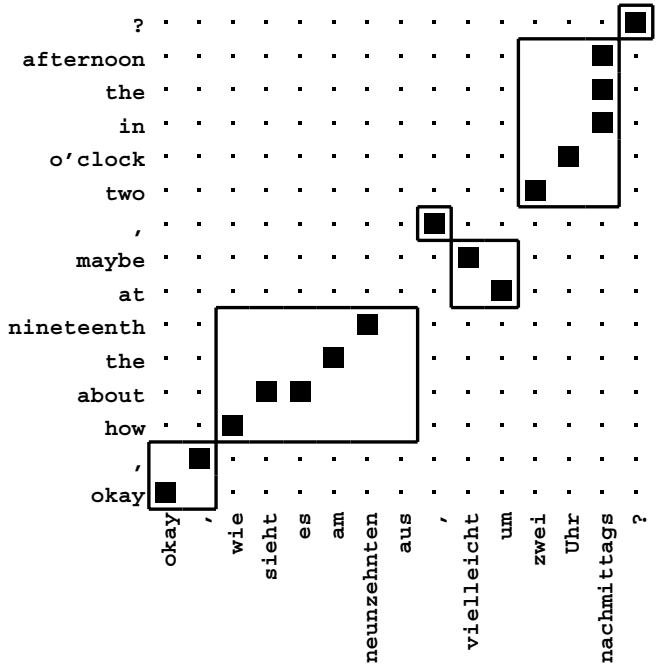

FIGURE 11.5
Example of alignment templates for a German-English sentence pair.

these examples, the advantage of the alignment template is that both the word context and the local re-ordering of words can be taken into account.

To describe the alignment template approach in a formal way, we first decompose both the source sentence f_1^J and the target sentence e_1^I into a sequence of word groups:

$$f_1^J = \tilde{f}_1^K , \qquad \tilde{f}_k = f_{j_{k-1}+1}, ..., f_{j_k} , \quad k = 1, ..., K$$
$$e_1^I = \tilde{e}_1^K , \qquad \tilde{e}_k = e_{i_{k-1}+1}, ..., e_{i_k} , \quad k = 1, ..., K$$

To simplify the notation and the presentation, we ignore the fact that there can be a large number of possible segmentations, and assume that there is only one segmentation. We distinguish two levels of alignments: alignment *within* the word groups and alignment *between* word groups. For the alignment $\tilde{a}_1^K$ *between* the source word

groups $\tilde{f}_1^K$ and the target word groups $\tilde{e}_1^K$, we have the following equation:

$$\begin{aligned} Pr(f_1^J|e_1^I) &= Pr(\tilde{f}_1^K|\tilde{e}_1^K) \\ &= \sum_{\tilde{a}_1^K} Pr(\tilde{a}_1^K, \tilde{f}_1^K|\tilde{e}_1^K) \\ &= \sum_{\tilde{a}_1^K} Pr(\tilde{a}_1^K|\tilde{e}_1^K) \cdot Pr(\tilde{f}_1^K|\tilde{a}_1^K, \tilde{e}_1^K) \\ &= \sum_{\tilde{a}_1^K} \prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_{k-1}, K) \cdot p(\tilde{f}_k|\tilde{e}_{\tilde{a}_k}) \end{aligned}$$

where we have used a first-order alignment model $p(\tilde{a}_k|\tilde{a}_{k-1}, K)$.
For the alignment *within* the word group, we introduce a new hidden variable z, which will be referred to as alignment template. Denoting the source word group by $\tilde{f} = \tilde{f}_1^{J'}$ and the target word group by $\tilde{e} = \tilde{e}_1^{I'}$, we have the model:

$$p(\tilde{f}|\tilde{e}) = \sum_z p(z|\tilde{e}) \cdot p(\tilde{f}|z, \tilde{e})$$

Each alignment template z can be represented as a binary matrix with I' rows and J' columns where only the high-probability (i, j)-pairs have a value of $z_{ij} = 1$ (denoted by a full square in Figure 11.5). The probabilities $p(z|\tilde{e})$ and $p(\tilde{f}|z, \tilde{e})$ are determined using the aligned training corpus and are set to zero if the triple $(\tilde{f}, \tilde{e}, z)$ did not occur in the training corpus. If the triple did occur in the training corpus, we use the following model for $p(\tilde{f}|z, \tilde{e})$:

$$\begin{aligned} p(\tilde{f}|z, \tilde{e}) &= \prod_{j=1}^{J'} \sum_{i=1}^{I'} p(i|j, z) \cdot p(\tilde{f}_j|\tilde{e}_i) \\ p(i|j, z) &= \frac{z_{ij}}{\sum_{i'} z_{i'j}} \end{aligned}$$

Typically, there is an additional refinement step by introducing a set of bilingual word classes that are determined automatically [31]. The alignment templates are then defined at this level of bilingual word classes rather than on the level of the words themselves. This step slightly improves the generalization capability for unseen test data.

11.4.2 Training

The training of alignment templates starts with the training of two HMM alignment models for *each* of the two translation directions (source → target and target → source). As a result, we obtain an alignment matrix for each training sentence pair by merging the alignment paths of both translation directions. In such an alignment matrix, it is possible that one source word is aligned to more than one target word.

Using the whole set of alignment matrices for the training corpus, we then extract the alignment templates by considering all possible source-target word groups under the constraint that the words within the source/target phrase are only aligned to words within the target/source phrase. The probability $p(z|\tilde{e})$ is then estimated as the relative frequency for the event pairs $(z, \tilde{e})$.

11.4.3 Search

To perform the search, we use the following models:

- As language model, we use a class-based n–gram (e.g. trigram or 5-gram) language model with backing-off. Typically, this is slightly better than the standard bigram language model.
- We assume that all possible segmentations have the same probability.
- The alignment model at the template level is an HMM-type alignment model. Obviously, as usual, all words in the source string must be covered.

To generate the unknown target sentence in the search procedure, we have to allow for all possible segmentations of the source sentence into word groups, for all possible alignments *between the word groups* and for possible alignments *within* the word groups. There are a couple of simplifications and approximations to reduce the computational cost of the search, which cannot be described here for space limitations. In principle, we use a beam search strategy. The search algorithm builds up hypotheses of increasing length along the positions of the target string. During the search process, we compute an estimate for the remaining portion of the source string that has not been covered. This estimate for the remaining portion is combined with the probability score for the already covered portion of the source string to narrow down the search to the most promising search hypotheses.

11.5 Experimental Results

11.5.1 The Task and the Corpus

The goal of the VERBMOBIL [49] is the translation of spoken dialogues in the domains of appointment scheduling and travel planning. In a typical situation, a native German speaker and a native English speaker conduct a dialogue where they can only interact by speaking and listening to the VERBMOBIL system.
Within the VERBMOBIL project, spoken dialogues were recorded. These dialogues were manually transcribed and later manually translated by VERBMOBIL partners (Hildesheim for Phase I and Tübingen for Phase II). Since different human translators were involved, there is great variability in the translations.

Each of these so-called dialogue turns may consist of several sentences spoken by the same speaker and is sometimes rather long. As a result, there is no one-to-one correspondence between source and target sentences. To achieve a one-to-one correspondence, the dialogue turns are split into shorter segments using punctuation marks as potential split points. Since the punctuation marks in source and target sentences are not necessarily identical, a dynamic programming approach is used to find the optimal segmentation points. The number of segments in the source sentence and in the test sentence can be different. The segmentation is scored using a word-based alignment model, and the segmentation with the best score is selected. This segmented corpus is the starting point for the training of translation and language models. Alignment models of increasing complexity are trained on this bilingual corpus [32, 33].
A standard vocabulary had been defined for the various speech recognizers used in VERBMOBIL. However, not all words of this vocabulary were observed in the training corpus. Therefore, the translation vocabulary was extended semi-automatically by adding about 13 000 German–English word pairs from an online bilingual lexicon available on the web. The resulting lexicon contained not only word-word entries, but also multi-word translations, especially for the large number of German compound words. To counteract the sparseness of the training data, a couple of straightforward rule-based preprocessing steps were applied *before* any other type of processing:

- categorization of proper names for persons and cities,
- normalization of:
  – numbers,
  – time and date phrases,
  – spelling: `don't` $\rightarrow$ `do not`,...
- splitting of
  German compound words.

Table 11.1 gives the characteristics of the training corpus and the lexicon. The 58 000 sentence pairs comprise about half a million running words for each language of the bilingual training corpus. The vocabulary size is the number of distinct full-form words seen in the training corpus. Punctuation marks are treated as regular words in the translation approach. Notice the large number of word singletons, i. e. words seen only once. The extended vocabulary is the vocabulary after adding the manual bilingual lexicon.

11.5.2 Offline Results

During the progress of the VERBMOBIL project, different variants of statistical translation were implemented, and experimental tests were performed for both text and speech input. To summarize these experimental tests, we briefly report experimental offline results for the following translation approaches:

- single-word based approach [44];
- alignment template approach [34];

- cascaded transducer approach [47]:
  Unlike the other two-approaches, this approach requires a semi-automatic training procedure, in which the structure of the finite state transducers is designed manually. For more details, see [47].

The offline tests were performed on text input for the translation direction from German to English. The test set consisted of 251 sentences, which comprised 2197 words and 430 punctuation marks. The results are shown in Table 11.2. To judge and compare the quality of different translation approaches in offline tests, we typically use the following error measures [29]:

- mWER (multi-reference word error rate):
  For each test sentence in the source language, there are *several* reference translations in the target language. For each translation of the test sentence, the edit distances (number of substitutions, deletions, and insertions as in speech recognition) to all reference translations are calculated, and the smallest distance is selected and used as error measure. For an extension of mWER-based measures, see also [36].

- SSER (subjective sentence error rate) [29]:
  Each translated sentence is judged by a human examiner according to an error scale from 0.0 (semantically and syntactically correct) to 1.0 (completely wrong).

Both error measures are reported in Table 11.2. Although the experiments with the cascaded transducers [47] were not fully optimized yet, the preliminary results indicated that this semi-automatic approach does not generalize as well as the other two fully automatic approaches. Among these two, the alignment template approach was found to work consistently better across different test sets (and also tasks different from VERBMOBIL). Therefore, the alignment template approach was used in the final VERBMOBIL prototype system.

TABLE 11.1
Bilingual training corpus, recognition lexicon and translation lexicon.

		German	English
Training Text	Sentence Pairs	58 073	
	Words	418 979	453 632
	Words + Punct.Marks	519 523	549 921
	Vocabulary	7 940	4 673
	Singletons	44.8%	37.6%
Recognition	Vocabulary	10 157	6 871
Translation	Added Word Pairs	12 779	
	Vocabulary	11 501	6 867

TABLE 11.2
Comparison of three statistical translation approaches (test on text input: 251 sentences = 2197 words + 430 punctuation marks).

Translation Approach	mWER [%]	SSER [%]
Single-Word Based	38.2	35.7
Alignment Template	36.0	29.0
Cascaded Transducers	>40.0	>40.0

11.5.3 Integration into the VERBMOBIL Prototype System

The statistical approach to machine translation is embodied in the *stattrans* module which is integrated into the VERBMOBIL prototype system. We briefly review those aspects of it that are relevant for the statistical translation approach. The implementation supports the translation directions from German to English and from English to German. In regular processing mode, the *stattrans* module receives its input from the *repair* module [41]. At that time, the word lattices and best hypotheses from the speech recognition systems have already been prosodically annotated, i.e. information about prosodic segment boundaries, sentence mode, and accentuated syllables are added to each edge in the word lattice [5]. The translation is performed on the single best sentence hypothesis of the recognizer.
The prosodic boundaries and the sentence mode information are utilized by the *stattrans* module as follows. If there is a major phrase boundary, a full stop or question mark is inserted into the word sequence, depending on the sentence mode as indicated by the *prosody* module. Additional commas are inserted for other types of segment boundaries. The *prosody* module calculates probabilities for segment boundaries, and thresholds are used to decide if the sentence marks are to be inserted. These thresholds have been selected in such a way that, on the average, for each dialogue turn, a good segmentation is obtained. The segment boundaries restrict possible word reordering between source and target language. This not only improves translation quality, but also restricts the search space and thereby speeds up the translation process.

11.5.4 Final Evaluation

Whereas the offline tests reported above were important for the optimization and tuning of the system, the most important evaluation was the final evaluation of the VERBMOBIL prototype in spring 2000. This final evaluation of the VERBMOBIL system was performed at the University of Hamburg [43].
Three other translation approaches had been integrated into the VERBMOBIL prototype system:

- a classical transfer approach [6, 18, 45],
  which is based on a manually designed analysis grammar, a set of transfer

rules, and a generation grammar,

- a dialogue act based approach [38],
  which amounts to a sort of *slot filling* by classifying each sentence into one out of a small number of possible sentence patterns and filling in the slot values, and

- an example-based approach [3],
  where a sort of nearest neighbor concept is applied to the set of bilingual training sentence pairs after suitable preprocessing.

In the final evaluation, human evaluators judged the translation quality for each of the four translation results using the following criterion:

Is the sentence approximatively correct: yes/no?

The evaluators were asked to pay particular attention to the semantic information (e.g. date and place of meeting, participants etc.) contained in the translation. A missing translation as it may happen for the transfer approach or other approaches was counted as wrong translation. The evaluation was based on 5069 dialogue turns for the translation from German to English and on 4136 dialogue turns for the translation from English to German. The speech recognizers used had a word error rate of about 25%. The overall sentence error rates, i.e. resulting from recognition *and* translation, are summarized in Table 11.3. As we can see, the error rates for the statistical approach are smaller by a factor of about 2 in comparison with the other approaches. Although the absolute values of the error rates shown in Table 11.3 may depend heavily on the specific test conditions used in [43], there is no reason to assume that the *relative* performance of the four approaches will be thereby changed. In addition to the four methods shown in Table 11.3, there was a fifth method called substring-based translation [10]. This method is based on bilingual word strings that are similar to alignment templates and are extracted from statistical alignments. Its performance was slightly inferior to the statistical approach. However, this method was not part of the original prototype system and was not evaluated on exactly the same corpus [43].

TABLE 11.3
Sentence error rates of final evaluation (speech recognizer with WER=25%; corpus of 5069 and 4136 dialogue turns for translation German to English and English to German, respectively).

Translation Method	Error [%]
Semantic Transfer	62
Dialogue Act Based	60
Example Based	52
Statistical	29

In agreement with other evaluation experiments, these experiments show that the statistical modelling approach may be comparable to or better than the conventional rule-based approach. In particular, the statistical approach seems to have the advantage if robustness is important, e.g. when the input string is not grammatically correct or when it is corrupted by recognition errors.

Although both text and speech input are translated with good quality on the average by the statistical approach, there are examples where the syntactic structure of the produced sentence is not correct. Some of these syntactic errors are related to long range dependences and syntactic structures that are not captured by the n-gram language model used. To cope with these problems, morpho-syntactic analysis [28] and grammar-based language models [40] are currently being studied.

11.6 Speech Translation: The Integrated Approach

11.6.1 Principle

In the Bayes decision rule, we have so far assumed written input, i.e. perfect input with no errors. When trying to derive a strict statistical decision rule for translation of spoken input, we are faced with the additional complication of speech recognition errors. So the question comes up of how to integrate the probabilities of the speech recognition process into the translation process. Although there have been activities in speech translation at several places [1, 25, 46], there has been not much work on this question of recognition/translation integration.

Considering the problem of speech input rather than text input for translation, we can distinguish three levels, namely the acoustic vectors $x_1^T = x_1 ... x_t ... x_T$ over time $t = 1...T$, the source words f_1^J, and the target words e_1^I:

$$x_1^T \rightarrow f_1^J \rightarrow e_1^I$$

From a strict point of view, the source words f_1^J are not of direct interest for the speech translation task. Mathematically, this is captured by introducing the possible source word strings f_1^J as hidden variables into the Bayes decision rule:

$$
\begin{aligned}
&\arg\max_{e_1^I} Pr(e_1^I|x_1^T) \;=\\
&= \arg\max_{e_1^I} \left\{ Pr(e_1^I) \cdot Pr(x_1^T|e_1^I) \right\} \\
&= \arg\max_{e_1^I} \left\{ Pr(e_1^I) \cdot \sum_{f_1^J} Pr(f_1^J, x_1^T|e_1^I) \right\} \\
&= \arg\max_{e_1^I} \left\{ Pr(e_1^I) \cdot \sum_{f_1^J} Pr(f_1^J|e_1^I) \cdot Pr(x_1^T|f_1^J, e_1^I) \right\} \\
&= \arg\max_{e_1^I} \left\{ Pr(e_1^I) \cdot \sum_{f_1^J} Pr(f_1^J|e_1^I) \cdot Pr(x_1^T|f_1^J) \right\} \\
&\cong \arg\max_{e_1^I} \left\{ Pr(e_1^I) \cdot \max_{f_1^J} \Big\{ Pr(f_1^J|e_1^I) \cdot Pr(x_1^T|f_1^J) \Big\} \right\}
\end{aligned}
$$

Here, we have made no special modelling assumption, apart from the reasonable assumption that

$$Pr(x_1^T|f_1^J, e_1^I) = Pr(x_1^T|f_1^J) \;,$$

i.e. the target string e_1^I does not help to predict the acoustic vectors (in the source language) *if* the source string f_1^J is given. In addition, in the last equation, we have used the maximum approximation. Only in that special case of speech translation, at least from a strict point of view, there is the notion of a 'recognized' source word sequence f_1^J. However, this word sequence is very much determined by the combination of the language model $Pr(e_1^I)$ of the target language and the translation model $Pr(f_1^J|e_1^I)$. In contrast, in recognition, there would be only the language model $Pr(f_1^J)$.

11.6.2 Practical Implementation

When presenting the statistical approach to written language translation, the tacit assumption had been that the source sentence f_1^J was well formed. However, for speech input, this assumption is no more valid. Therefore, to take into account the requirement of 'well-formedness', we use a more complex translation model by including the dependence on the predecessor word:

$$p(f_j|f_{j-1}, e_{a_j}) \quad \text{in lieu of} \quad p(f_j|e_{a_j})$$

$$Pr(f_1^J|e_1^I) = \sum_{a_1^J} \prod_j \left[p(a_j|a_{j-1}, I) \cdot p(f_j|f_{j-1}, e_{a_j}) \right]$$

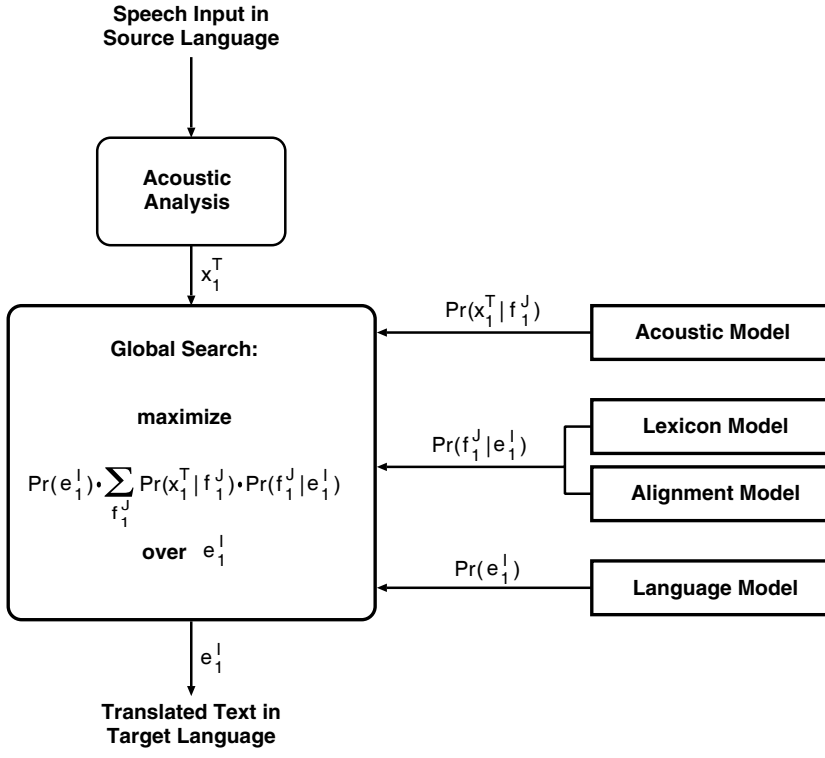

FIGURE 11.6
Integrated architecture of speech translation approach based on Bayes decision rule.

For the sake of simplicity, here we have chosen the bigram dependence.

It is instructive to re-interpret already existing approaches for handling speech input in a translation task in the light of the Bayes decision rule for speech translation, even if these approaches are not based on stochastic modelling. The key issue in all these approaches is the question of how the requirement of having both a well-formed source sentence f_1^J and a well-formed target sentence e_1^I at the same time is satisfied. From the statistical point of view, this question is captured by finding suitable models for the *joint* probability $Pr(f_1^J, e_1^I) = Pr(e_1^I) \cdot Pr(f_1^J | e_1^I)$.

From the decision rule, it is clear that the translation process will have an effect on the recognition process only if the target language model $Pr(e_1^I)$ is sufficiently strong or, to be more exact, if its strength is comparable to that of the source language model $Pr(f_1^J)$. We mention the following approaches:

- In many systems, the method of n-best lists is used. The recognizer produces a list of n best source sentences, and the translation system works as a filter that selects one out of the n sentences using some suitable criterion. This joint

generation and filtering process can be viewed as a crude approximation of the joint probability $Pr(f_1^J, e_1^I)$.

- When using finite-state methodology rather than a fully stochastic approach, the probability $Pr(f_1^J, e_1^I)$ is modelled by the finite-state network of the corresponding transducer, which is typically refined by domain and range restrictions [14, 46].

- In the extreme case, we might be only interested in the *meaning* of the target translation. Such an approach was used in [38] for the Verbmobil task. In Bayes decision rule, this case is captured by putting most emphasis on a *semantically* constrained language model $Pr(e_1^I)$. In addition, confidence measures [51] can be used to filter out those words that are most likely to have been recognized correctly.

However, it is clear that none of these approaches fully implements the integrated coupling of recognition and translation from a statistical point of view. We consider this integrated approach and its suitable implementation to be an open question for future research on spoken language translation.
What we have considered here is speech input in the *source* language. In machine-aided translation, the speech input is in the *target* language, and thus the source sentence is typically used to change the language model in the target language [11, 17].

11.7 Summary

In this chapter, we have given an overview of the statistical approach to machine translation and especially its implementation in the VERBMOBIL prototype system. The statistical system has been trained on about 500 000 running words from a bilingual German–English corpus. Translations are performed for both directions, i.e. from German to English and from English to German. Comparative evaluations with other translation approaches of the VERBMOBIL prototype system show that the statistical translation is superior, especially in the presence of speech input and ungrammatical input. In addition, we have presented the fully integrated approach to spoken language translation.

Acknowledgment

The authors would like to thank the researchers in ITI Valencia and Aachen University of Technology who participated in the projects VERBMOBIL and EUTRANS and developed the approaches presented in this chapter. In particular, the authors would like to thank Enrique Vidal and Francisco Casacuberta for many discussions.

The work reported here was partly carried out in the VERBMOBIL project (contract number 01 IV 701 T4) funded by the German Federal Ministry of Education, Science, Research, and Technology and in the EUTRANS project (contract number IT-LTR-OS 30268) funded by the European Union.

11.8 References

[1] H. Alshawi, F. Xiang: English-to-Mandarin speech translation with head transducers. *Spoken Language Translation Workshop, 35th Annual Conf. of the Assoc. for Computational Linguistics*, pp. 54–60, Madrid, Spain, July 1997.

[2] H. Alshawi, S. Bangalore, S. Douglas: Learning dependency translation models as collection of finite-state head transducers. *Computational Linguistics*, Vol. 26, No. 1, pp. 45–60, 2000.

[3] M. Auerswald: Example-based machine translation with templates. In [49], pp. 418–427.

[4] S. Bangalore, G. Riccardi: Finite-state models for lexical reordering in spoken language translation. *Int. Conf. on Spoken Language Processing,* Vol. IV, pp. 422–425, Bejing, China, Oct. 2000.

[5] A. Batliner, J. Buckow, H. Niemann, E. Nöth, V. Warnke: The prosody module. In [49], pp. 106–121.

[6] T. Becker, A. Kilger, P. Lopez, P. Poller: The Verbmobil generation component VM-GECO. In [49], pp. 481–496.

[7] A. L. Berger, P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, J. D. Lafferty, R. L. Mercer, H. Printz, L. Ures: The Candide system for machine translation. *ARPA Human Language Technology Workshop*, Plainsboro, NJ, Morgan Kaufmann Publishers, pp. 152–157, San Mateo, CA, March 1994.

[8] A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, A. S. Kehler: Language translation apparatus and method of using context-based translation models. *United States Patent*, Patent Number 5510981, April 1996.

[9] A. L. Berger, S. Della Pietra, V. Della Pietra: A maximum entropy approach to natural language processing. *Computational Linguistics*, Vol. 22, No. 1, pp. 39–71, 1996.

[10] U. Block: Example-based incremental synchronous interpretation. In [49], pp. 411–417.

[11] P. F. Brown, S. F. Chen, S. A. Della Pietra, V. J. Della Pietra, A. S. Kehler, R. L. Mercer: Automatic speech recognition in machine-aided translation. *Computer Speech and Language*, Vol. 8, pp. 177–187, 1994.

[12] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer: The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, Vol. 19, No. 2, pp. 263–311, 1993.

[13] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, R. L. Mercer: Class–based n–gram models of natural language. *Computational Linguistics*, Vol. 18, No. 4, pp. 467–479,1992.

[14] F. Casacuberta: Finite-state transducers for speech input translation. *IEEE Automatic Speech Recognition and Understanding Workshop*, Madonna di Campiglio, Italy, 8 pages, CD ROM, IEEE Catalog No. 01EX544, Dec. 2001.

[15] N. Chomsky: Quine's empirical assumptions. p. 57 in D. Davidson, J. Hintikka (eds.): *Words and objections. Essays on the work of W. V. Quine*, Reidel, Dordrecht, The Netherlands, 1969.

[16] I. Dagan, K. Church, W. A. Gale: Robust bilingual word alignment for machine-aided translation. *Workshop on Very Large Corpora*, pp.1–8, Columbus, OH, June 1993.

[17] M. Dymetman, J. Brousseau, G. Foster, P. Isabelle, Y. Normandin, P. Plamondon: Towards an automatic dictation system for translators: the TransTalk project. *Int. Conf. on Spoken Language Processing,* Vol. II, pp. 691–694, Yokohama, Japan, Sep. 1994.

[18] M. C. Emele, M. Dorna, A. Lüdeling, H. Zinsmeister, C. Rohrer: Semantic-based transfer. In [49], pp. 359–376.

[19] EuTrans Project; Instituto Tecnológico de Informática (ITI Valencia, Spain), Fondazione Ugo Bordoni (FUB Rome, Italy), Lehrstuhl für Informatik VI (RWTH Aachen, Germany), Zeres GmbH (Bochum, Germany): Example-based language translation systems. *Final report of the EuTrans project* (EU project number 30268), July 2000.

[20] I. García-Varea, F. J. Och, H. Ney, F. Casacuberta: Refined lexicon models for statistical machine translation using a maximum entropy approach. *39th Annual Meeting of the Assoc. for Computational Linguistics,* pp. 204–211, Toulouse, France, July 2001.

[21] U. Germann, M. Jahr, K. Knight, D. Marcu, K. Yamada: Fast decoding and optimal decoding for machine translation. *39th Annual Meeting of the Assoc. for Computational Linguistics,* pp. 228–235, Toulouse, France, July 2001.

[22] F. Jelinek: *Statistical methods for speech recognition.* MIT Press, Cambridge, MA, 1997.

[23] R. Kneser, H. Ney: Improved clustering techniques for class–based statistical language modelling. *Europ. Conf. on Speech Communication and Technology,* pp. 973–976, Berlin, Germany, Sept. 1993.

[24] K. Knight: Decoding complexity in word-replacement translation models. *Computational Linguistics*, No. 4, Vol. 25, pp. 607–615, 1999.

[25] A. Lavie, L. Levin, A. Waibel, D. Gates, M. Gavalda, L. Mayfield: JANUS: Multi-lingual translation of spontaneous speech in a limited domain. *2nd Conf. of the Assoc. for Machine Translation in the Americas*, pp. 252-255, Montreal, Canada, Oct. 1995.

[26] H. Ney: Speech translation: coupling of recognition and translation. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing,* pp. I-517-520, Phoenix, AR, March 1999.

[27] H. Ney, S. Nießen, F. J. Och, H. Sawaf, C. Tillmann, S. Vogel: Algorithms for statistical translation of spoken language. *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 1, pp. 24–36, Jan. 2000.

[28] S. Nießen, H. Ney: Improving SMT quality with morpho-syntactic analysis. *18th Int. Conf. on Computational Linguistics*, pp. 1081–1085, Saarbrücken, Germany, July 2000.

[29] S. Nießen, F. J. Och, G. Leusch, H. Ney: An evaluation tool for machine translation: fast evaluation for MT research. *2nd Int. Conf. on Language Resources and Evaluation*, pp. 39–45, Athens, Greece, May 2000.

[30] S. Nießen, S. Vogel, H. Ney, C. Tillmann: A DP based search algorithm for statistical machine translation. *36th Annual Meeting of the Assoc. for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, pp. 960–967, Montreal, Canada, Aug. 1998.

[31] F. J. Och: An efficient method to determine bilingual word classes. *9th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics*, pp. 71–76, Bergen, Norway, June 1999.

[32] F. J. Och, H. Ney: A comparison of alignment models for statistical machine translation. *18th Int. Conf. on Computational Linguistics*, pp. 1086–1090, Saarbrücken, Germany, July 2000.

[33] F. J. Och, H. Ney: Improved statistical alignment models. *38th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 440-447, Hong Kong, Oct. 2000.

[34] F. J. Och, C. Tillmann, H. Ney: Improved alignment models for statistical machine translation. *Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20–28, University of Maryland, College Park, MD, June 1999.

[35] F. J. Och, N. Ueffing, H. Ney: An efficient A* search algorithm for statistical machine translation. *Data-Driven Machine Translation Workshop, 39th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 55–62, Toulouse, France, July 2001.

[36] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu: BLEU: a method for automatic evaluation of machine translation. *IBM Research Report*, 8 pages, Yorktown Heights, NY, Sept. 2001.

[37] L. R. Rabiner, B. H. Juang: *Fundamentals of speech recognition*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[38] N. Reithinger, R. Engel: Robust content extraction for translation and dialog processing. In [49], pp. 428–437.

[39] S. D. Richardson, W. B. Dolan, A. Menezes, M. Corston-Olivier: Overcoming the customization bottleneck using example-based MT. *Data-Driven Machine Translation Workshop, 39th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 9–16, Toulouse, France, July 2001.

[40] H. Sawaf, K. Schütz, H. Ney: On the use of grammar based language models for statistical machine translation. *6th Int. Workshop on Parsing Technologies*, pp. 231–241, Trento, Italy, Feb. 2000.

[41] J. Spilker, M. Klarner, G. Görz: Processing self-corrections in a speech-to-speech system. In [49], pp. 131–140.

[42] E. Sumita: Example-based machine translation using DP-matching between word sequences. *Data-Driven Machine Translation Workshop, 39th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 1–8, Toulouse, France, July 2001.

[43] L. Tessiore, W. V. Hahn: Functional validation of a machine translation system: Verbmobil. In [49], pp. 611–631.

[44] C. Tillmann, H. Ney: Word re-ordering in a DP-based approach to statistical MT. *18th Int. Conf. on Computational Linguistics 2000*, pp. 850–856, Saarbrücken, Germany, Aug. 2000.

[45] H. Uszkoreit, D. Flickinger, W. Kasper, I. A. Sag: Deep linguistic analysis with HPSG. In [49], pp. 216–263.

[46] E. Vidal: Finite-state speech-to-speech translation. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 111–114, Munich, Germany, April 1997.

[47] S. Vogel, H. Ney: Translation with cascaded finite-state transducers. *38th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 23–30, Hong Kong, Oct. 2000.

[48] S. Vogel, H. Ney, C. Tillmann: HMM-based word alignment in statistical translation. *16th Int. Conf. on Computational Linguistics*, pp. 836–841, Copenhagen, Denmark, August 1996.

[49] W. Wahlster (Ed.): *Verbmobil: Foundations of speech-to-speech translation.* Springer-Verlag, Berlin, Germany, 2000.

[50] Y.-Y. Wang, A. Waibel: Decoding algorithm in statistical translation. *35th Annual Conf. of the Assoc. for Computational Linguistics*, pp. 366–372, Madrid, Spain, July 1997.

[51] F. Wessel, R. Schlüter, K. Macherey, H. Ney: Confidence measures for large vocabulary continuous speech recognition. *IEEE Trans. on Speech and Audio Processing,* Vol. 9, No. 3, pp. 288–298, March 2001.

[52] D. Wu: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, Vol. 23, No. 3, pp. 377–403, 1997.

[53] K. Yamada, K. Knight: A syntax-based statistical translation model. *39th Annual Meeting of the Assoc. for Computational Linguistics,* pp. 523–530, Toulouse, France, July 2001.

12

Modeling Topics for Detection and Tracking

James Allan
University of Massachusetts Amherst

CONTENTS

Topic detection and tracking (TDT) is a research program and an evaluation paradigm that investigates techniques for automatically organizing broadcast news stories by the events that they describe. TDT is an outgrowth of information retrieval (IR) technology and shares many of its techniques and ideas. The specific tasks within event-based organization and the nature of the stories to which those tasks are applied, means that TDT admits a range of approaches that are not universally applicable within IR.

One of the research issues within TDT is representing a news topic based on sparse information—for example, based on a single story. This chapter discusses several ways in which topics are modeled within the TDT research community and compares and contrasts them. We are agnostic as to which is the best model—research suggests that most of the approaches are equally effective. However, it is disappointing that no techniques explicitly model the events out of which TDT topics arise. We will conclude the chapter with speculation about how events might be more directly incorporated into the topic models.

12.1 Topic Detection and Tracking

The goal of TDT research is to organize news stories by the events that they describe, and to do that as soon as the stories appear, whether as newswire, television, or radio.

That is, the decision about how to handle a story must be made before any additional stories are processed.

The TDT research program began in 1996 as a collaboration between Carnegie Mellon University, Dragon Systems, the University of Massachusetts, and DARPA [2]. That group of researchers ran a pilot study that defined the basic tasks of TDT and how they should be evaluated. To find out how well classic IR technologies addressed TDT, they created a small collection of news stories and identified some topics within them.

Because the results were encouraging, a larger and more formal series of evaluations were held every year from 1998 through 2001 (and continues in 2002 and perhaps longer). These evaluations more carefully defined the notions of topic and event, developed the set of tasks more fully, and constructed a larger and richer corpus of stories and topics.

12.1.1 Topic and Events

An *event* was defined as something that happens at some specific time and place, along with all necessary preconditions and unavoidable consequences. That is, it is something that happens in the real world. A particular earthquake is an event, as is the discovery of a new comet.

A *topic* is meant to capture the larger set of happenings that are related to some triggering event. The official definition of a topic is that it is a seminal event, along with all directly related events and activities. By forcing the additional events to be directly related, the topic is prevented from spreading out to include too much news. One way to think of a topic is that, given a starting event, it includes the additional events that a typical reader would expect to see in follow-up news. (Or, given an event that is not of interest, the set of following events that the reader would prefer *not* to see.)

Topics, then, are anchored in time and space by the seminal event. The focus on events and the tight relation to time distinguish TDT topics from the more general use of the word "topic" within information retrieval. In that setting, a topic is usually subject-based: it represents an area of interest to the searcher. Some subject-based topics are identical to TDT topics ("I am interested in information about the Kobe earthquake of 1995") but some have no parallel in TDT ("Tell me about endangered species in Africa").

The differences between TDT topics and IR topics means that different techniques should be useful to address their respective tasks. The similarities are sufficient enough, though, that most research has focused on the direct application of IR methods to TDT tasks.

12.1.2 TDT Tasks

The TDT evaluation program defines five tasks for organizing news by events: segmentation, cluster detection, tracking, new event detection, and link detection.

12.1.2.1 Segmentation

News that arrives via newswire is divided into individual stories, whereas television and radio news is not. The task of segmentation is to break an audio track into discrete stories, each on a single topic. Most research on this problem has used speech recognizer output as a starting point rather than working on the audio itself.

This task is a necessary precondition to all of the other tasks since they assume a set of stories that need to be organized. We will not talk about segmentation in the rest of this chapter. It has been discussed elsewhere [49, 10]

12.1.2.2 Cluster Detection

In the cluster detection task (also referred to simply as "detection"), a system must place all arriving news stories into groups based on their topics. If no existing group's topic matches the story sufficiently, the system must decide whether to create a new group. The decision about how to process an individual story must be made before the next story is considered.

The TDT evaluation program requires that the each story be placed in precisely one cluster, implying that each story is about a single topic. This simplifying assumption was useful in early evaluations and is being dropped after TDT 2001.

12.1.2.3 Tracking

The task of tracking starts with a small set of news stories that a user has identified as being on the same topic. Given that set, the system must monitor the stream of arriving news to find all additional stories on the same topic. As with detection, a decision must be made about each story before additional stories can be seen. Further, the system is never given feedback (other than a final evaluation) about whether it has made a correct decision.

This task is analogous to the IR filtering problem [38]. Differences between the two lie primarily in the definition of topic (as in Section 12.1.1) and in different evaluation paradigms [26, 7, 5].

12.1.2.4 New Event Detection

The task of new event detection focuses on the cluster creation aspect of cluster detection. A system is evaluated entirely on its ability to decide when a new topic (event) appears. Whether or not the remaining stories in the topic are properly placed in their topics is unimportant. As usual, decisions must be made as stories arrive. This task is also referred to as "first story detection."

12.1.2.5 Link Detection

The final TDT task, link detection, was created as a core technology for the other tasks. The idea is to determine whether or not two randomly presented stories discuss the same topic. A solution to this task could be used to solve new event detection,

for example, by comparing the newly arrived story to every story in the past. If no earlier story had the same topic, then a new topic can be declared.
It is not necessary that link detection be the technology to address other tasks, but most approaches to TDT problems use ideas similar to this task.

12.1.3 Corpora

The corpus for the pilot study included about 16,000 stories from two sources (CNN and Reuters) gathered from the last half of 1994 and the first half of 1995. In addition, 25 topics were selected by the researchers that each story was judged against. The construction of the corpus and the formation of the topics was sufficient for a pilot study, but not for a more rigorous evaluation.
For the remaining TDT evaluations, the Linguistic Data Consortium was contracted to create the corpora, topics, and judgments [14]. Three corpora have been created to date. The corpora contain substantially more news stories than in the pilot study, include stories created from audio sources, and incorporate news written or read in foreign languages.
The TDT-2 corpus (the pilot corpus can be thought of as TDT-1) includes about 80,000 news stories from January through June of 1998. The stories comes from six English sources, three Chinese sources, and in 2002 is being augmented with some Arabic news from the same time period. Approximately 100 topics were identified by random selection of stories from the corpus and were judged against the entire set of 80,000 stories.* The TDT-2 collection was used as the evaluation collection for the 1998 evaluation and has been used as training data since then.
The TDT-3 corpus was created for the 1999 evaluation, but was also used for the 2000 and 2001 evaluations. It includes about 40,000 stories from the last three months of 1998, including eight English sources and three Chinese sources. Stories from four Arabic sources are being added during 2002. A total of 120 news topics were developed for this corpus. The first 60 were developed for the 1999 evaluation and have the unusual requirement that there must be at least four on-topic stories in *each* of English and Chinese. The other 60 topics were developed for the 2000 evaluation and remove that requirement—however, they were seeded equally from English and Chinese stories, so each language is represented. This latter set of topics is being judged against the newly added Arabic sources. In addition, the judgments for the second set of 60 topics were not done by complete reviewing of every story for each topic. Instead, as a cost saving measure, the annotations were made using human-guided search techniques (experiments showed equal accuracy between the approaches). In the 2001 evaluation, 30 topics from each set were chosen for the evaluation. The TDT-3 corpus will be used as training data for the TDT 2002 evaluation.
The latest TDT corpus is TDT-4, being created in the fall of 2001 and spring of 2002. It includes approximately 45,000 stories covering October 2000 through Jan-

*This is in contrast to typical IR judgments that are only judged against stories retrieved by some system participating in the evaluation. The reason that TDT can manage the complete judgment set is because the corpus is substantially smaller than a typical IR collection.

uary 2001. The multi-lingual aspect of TDT is being stressed more in this corpus, so the stories come from eight English sources, seven Chinese sources, and four Arabic sources. (Additional languages were collected in parallel but are not being included with the TDT-4 corpus at this time.) Sixty new topics are being developed from this data using a model similar to that of the second sixty topics in TDT-3 (i.e., seeded equally from each language). This corpus will be used for the TDT 2002 evaluation. In all three corpora, audio sources were passed through a speech recognition system and the output is included in the corpus. In addition, a reference closed-caption-quality transcript was made if it was not available (e.g., for radio sources). Non-English sources were recognized for audio and then translated to English using the SYSTRAN[†] system. The source language and the translation were made available to all sites.

12.1.4 Evaluation

All TDT tasks are envisioned as "on-line" tasks that must completely process each story before receiving any additional stories. Decisions are irrevocable, even if a mistake is detected later. This approach models a situation where the output is consumed immediately and in a time-critical fashion. It explores the core technology rather than how it might be used in an interactive setting [26]. Output is couched in terms of a detection task, where "yes" or "no" decisions must be made [19]. Evaluation is in terms of errors (misses and false alarms) and the tradeoff between them. Figure 12.1 shows a sample detection error tradeoff (DET) graph [28] for a TDT task. The false alarm rate is shown on the X-axis and the miss rate is on the Y-axis. As with most language tasks, the graph shows that the errors tradeoff against each other: lowering one tends to raise the other.

The official evaluation measure of TDT is based on a cost function, a weighted combination of miss and false alarm rates:

$$\text{Cost} = C_{\text{miss}} P(\text{miss}) P(\text{target}) + C_{\text{fa}} P(\text{fa}) P(\text{off-target})$$

where $P(\text{target})$ is the prior probability that a story will be on topic, C_x are user-specified values that reflect the cost associated with each error, and $P(\text{miss})$ and $P(\text{fa})$ are the actual system error rates. Within TDT evaluations, $C_{\text{miss}} = 10$, $C_{\text{fa}} = 1$, and $P(\text{target}) = 1 - P(\text{off-target}) = 0.02$ (derived from training data).

In fact, a normalized version of the cost function is used. A system that always answers "no" would have no false alarms, though it would have a 100% miss rate. That system would get a score of 0.2 (10×0.02). Similarly, a system that always answers "yes" would get a score of 0.98. To ensure that systems that under-perform such simple approaches are visible, the cost value is divided by the minimum of the "always say yes" or "always say no" approaches; in this case, by 0.2. A normalized detection cost of 1.0 means that the system performs exactly as well as a system that does no work.

[†] http://www.systransoft.com/

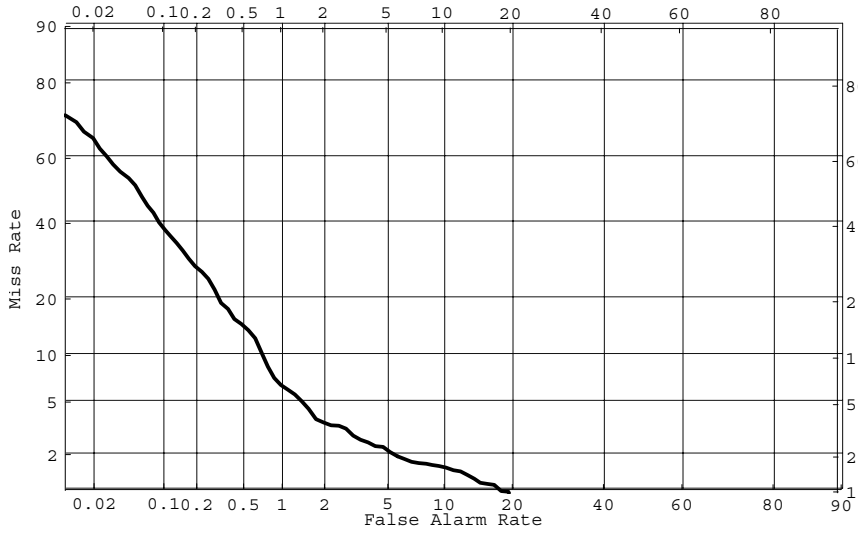

FIGURE 12.1
A sample detection error tradeoff (DET) curve for the TDT tracking task with one training story ($N_t = 1$).

Note that every point of the DET curve corresponds to a miss and false alarm rate, so there is a cost at every point of the curve. Within TDT, sites are expected to find a minimum cost on the curve, but differences between training and test data generally mean they miss it slightly. A common evaluation within the TDT community is the *minimum* cost that could have been attained using that DET curve. That is, an evaluation that sidesteps the final selection of threshold to get a sense of potential for a technology if threshold selection can be resolved.

All evaluations in TDT have been carried out by the National Institute of Standards and Technology. Participating sites were provided with the corpus and information that specified the starting condition for each task. For example, tracking required the set of training stories for each topic and the other tasks required a list of which stories to consider in the stream. Each site generated its decisions on the stories in the evaluation set and submitted them to NIST. In turn, NIST did the evaluation and generated comparative results of all the systems [19].

12.2 Basic Topic Models

Underlying all approaches to all of the TDT tasks is the notion of "topic." To address the tasks, it is necessary that a site somehow model topics and possibly the events within them. The model could be very simple (a list of significant words) or extremely complicated (in the spirit of a knowledge base of participants and their actions). Most work within TDT to date has represented topics either as a vector of weighted words or as a probability distribution of words. The approaches are similar in their implementation and effectiveness, but quite differently motivated.

12.2.1 Vector Space

The vector space has a long history within information retrieval research [39, 45] and is probably the most popular way of implementing an IR system. Not surprisingly, given the similarity in the problems, the vector space has been used by several sites in TDT [8, 32, 49, 17, 40].

The basic idea of the vector space approach is to represent items (stories or topics) as vectors in a high dimensional space. The dimensions correspond to the features that are used to represent the items and are orthogonal. Items that are similar enough—generally as measured either by the cosine of the angle between them or by their separation in Euclidean space—are assumed to be on the same topic.

Generally the words that occur in stories are the features of the vector space (it is obvious that the words are not independent of each other, but the model has been repeatedly shown to work empirically nonetheless). Words are given weights as discussed later in this chapter. The most common comparison function is the cosine of the angle between the two vectors:

$$\frac{\vec{u} \cdot \vec{v}}{||\vec{u}||_2 \cdot ||\vec{v}||_2}$$

When the vectors are normalized to length one, the cosine can be calculated just by taking the inner product of the two vectors (i.e., the denominator is 1.0).

A topic is modeled as one or more vectors in this model. When a set of stories is known to belong to a topic, the story vectors might be added to create a topic vector, perhaps with the more recent stories given higher weight. Some systems leave the vectors separate, noting that they are all part of the same topic, but keeping the topic model disperse [8].

Because the vector space model is so simple, the bulk of research is empirical efforts to find the right set of features, weights, and comparison method. The theory does not, in and of itself, provide much help in those efforts. However, the model remains popular precisely because it is so simple: it can be easily understood and implemented.

12.2.2 Language Models

Statistical language modeling approaches came to TDT via the speech recognition community [47] and the IR community [36, 35] (which also got the idea from speech recognition). In this approach, a topic is represented as a probability distribution of words. Higher probability words are much more likely to appear in on-topic stories than are lower probability words.

The difficult aspect of language modeling is coming up with good ways for estimating the probabilities. One or more stories that are known to be on the same topic are the starting point for building a topic model. The initial probability estimates come from the maximum likelihood estimate based on that document:

$$P_{\text{ml}}(w) = \frac{\text{tf}_w}{\text{tf}_*}$$

where tf_w represents the count of times that the word w occurs in a story, and tf_* is the total number of words in the story. This estimate is not sufficient because it will give zero probabilities to any word not in the story. For that reason, the maximum likelihood estimate is usually smoothed with estimates from a larger corpus of news stories with some mixing parameter λ that determines whether the story or the background corpus contributes more of the estimate.

There are generally two ways to use these topic models. The first is to see how likely it is that a particular story could be generated by the model, $P(\text{story}|M)$. A standard independence assumption is made and the probability is estimated as,

$$P(\text{story}|M) = \prod_{w \in \text{story}} P(w|M)$$

Stories that have higher probability are more likely to be part of the same topic that is modeled.

A second way to use topic models is to compare them directly. That is usually accomplished with a symmetric version of the Kullback-Leibler divergence [27], such as $D(M_1 \parallel M_2) + D(M_2 \parallel M_1)$. Other ways of comparing the models are also used [22].

12.3 Implementing the Models

In this section, we discuss several of the techniques that have been used in TDT to improve the model. Most of these techniques are applicable in some to either of the major types of model, though some make sense in only one, or some have been tried in only one. We discuss the use of named entities, the use of query expansion ideas, story clustering, and the inclusion of a time decay factor.

12.3.1 Named Entities

News is usually about people, so it seems reasonable that their names could be treated specially, in a way that would improve the accuracy of TDT systems. Named entity extraction systems have achieved high levels of accuracy, both for good quality news text and for the output of a speech recognizer [29, 31]. That means that it is possible to extract with reasonable confidence the names of people and organizations from the TDT news stories.

A simple way to use named entities in the model is to treat them as a separate part of the model and then merge the parts. For example, names of people in two stories might be compared and contribute part of the similarity, while comparison of organization or place names might contribute additional amounts [17]. On the hand, a system might just boost the weight of any words in the stories that come from names, giving them a larger contribution to the similarity when the names are in common [3]. Unfortunately, although names clearly provide importance for similarity and increasing their weight can improve results slightly, there has been no strong success so far.

We know of no uses of named entities in a language modeling system. Instead, systems use the individual words of the names independently. It may be that doing so allows for partial matches for variant forms of names (e.g., at least one word of *President Bush* and *George Bush* will match), an important issue while name coreference [9] remains a difficult problem.

Given the importance of people, places, and dates to news reporting, it is disappointing that named entities have not yet found a place of prominence in topic models. It may be that the models are not yet sophisticated enough to improve upon simple word-based models. For example, errors in name extraction and comparison may be swamping the value that using names adds. As the models become more sophisticated and accurate, there may be more value in using names.

12.3.2 Document Expansion

When topic models are created from a single story, they suffer from extremely limited vocabulary. There are countless words that could be used in the topic and some of them that have a high likelihood of appearing. So both modeling approaches need a way of expanding the set of words that are included in the topic model.

Vector space systems generally use techniques based on query expansion [18, 11, 48] that have been highly successful in IR evaluations. For example, in the segmentation task, a possible segmentation boundary could be checked by comparing the models generated by text on either side. To improve the chance of vocabulary overlap, the text could be used as a query to retrieve a few dozen related stories and then the most frequently occurring words from those stories could be used for the comparison [37]. Similar approaches could be used to add words to stories for the other tasks [6].

Adding words to a language model is in some ways simpler and in other ways more complex. It is simpler because smoothing with the background model (as described above) creates non-zero probabilities for every word in the corpus and therefore

brings those words into the topic to some degree. However, it does not increase the probability of words that are related to the topic.
One statistical language modeling approach to capturing the related words is to use relevance models [23]. That technique assumes that strongly similar news stories arise out of topics that are either the same as or strongly related to the topic that is being modeled. Each retrieved story generates a model and all of the models are combined to create a topic model:

$$P(w|\mathcal{M}) = \prod_{M \in \mathcal{M}} P(M)P(w|M)$$

The final result is similar in implementation to query expansion, but is justified probabilistically so that important properties of the language models are preserved. Relevance models result in substantial improvements in the link detection task [22].

12.3.3 Clustering

Grouping stories together can improve the representation of a topic if the stories are really on the same topic. The group provides a larger statistical sample from which the model can be estimated. Since grouping, or clustering, is a fundamental aspect of TDT (the detection task is a clustering task), it is not surprising that much work has focused on ways to leverage the ideas to improve TDT effectiveness [33].
The most obvious situation where clustering occurs is in the tracking task where several news stories are indicated as being on the same topic. Since the system "knows" that the stories are on the same topic, it can find features that are common in those stories that are not common elsewhere. Although the same process can be done starting from a single story, it is much more reliable with several stories. This approach is common in the vector space model, where the topic might be represented by the average of the on-topic story vectors minus the average off-topic story vector.
Clustering is not a clearly defined operation within a language modeling context, though if two models are similar enough it might make sense to merge them. To date, we know of no sites that have attempted to mimic vector clustering in the language model context. Instead, the larger set of on-topic stories is used to provide better estimates of word probabilities within the topic (in much the same way that relevance models do, as described above in Section 12.3.2).
Although clustering makes sense in tracking, it can also be done for all of the other tasks. Detection, for example, is a clustering task and requires that the system generate groupings. A system can adapt its topic models by incorporating newly arrived stories into the cluster when they seem sufficiently close [34, 24]. This notion of adapting the topic model is similar to the adaptive filtering ideas that have recently been adopted by the IR filtering community [38].
Note that some sites have found better results by keeping the stories within a cluster entirely distinct—in a sense, they are not doing the clustering [8]. Instead, the topic is represented by a set of topics, one for each story that is believed to be part of the topic.

Other sites have used a cross between the two ideas, clustering stories within the topic when they are sufficiently similar, but keeping them apart when they are not [16]. This creates a notion of "microclusters" that has the potential to provide flexibility when a topic is multi-faceted. It is similar to work done in the IR community on keeping track of shifting user interests [1].

12.3.4 Time Decay

Several sites have observed that the likelihood that two stories discuss the same topic diminishes as the stories are further separated in time. It is possible to leverage this observation by creating a prior probability that two stories are relevant and then modifying that based on content. Or in a vector space model, the cosine similarity function can be changed so that it includes a time decay [4].

After a surge of interest in the use of a time decay within the TDT pilot study and 1998 evaluation, it has used by substantially fewer systems in recent TDT evaluations. We suspect that the quality of word-based matching has improved to the extent that the time decay no longer helps as much as it did. It may also be the case that the evaluation topics are not enough to each other that the time they are reported can be an important distinguishing characteristic. Imagine how much simpler it is to separate similar terrorist events (e.g., bombings) given the time that the attack was reported.

12.4 Comparing Models

Once topic models are built, regardless of the model, they need to be used. Within TDT, that means comparing a story to a model to see if the story is part of the topic, or possibly comparing two models to determine the chance that they represent the same topic. In this section we discuss a few of the comparison functions that have been used. The first is specific to the vector space model: nearest neighbor decisions. The second, the use of decision trees, is independent of the models discussed here so far. We also discuss direct model comparisons within the language modeling framework.

12.4.1 Nearest Neighbors

In the vector space model, a topic might be represented as a single vector. So when a TDT system is running, it would have a large set of vectors representing all of the topics seen to date. A newly arrived news story can also be represented by a vector and dropped into the same space. To determine whether or not that story is on any of the existing topics, we consider the distance (usually measured by the cosine of the angle between vectors) between the story's vector and the closest topic vector. If it is sufficiently small, the story is assumed to be part of the topic. If it falls outside a

specified distance, the story is likely to be the seed of a new topic and a new vector can be formed.

The approach listed above is essentially a k-nearest neighbor approach, where k is one. A story is assigned the topic of its single nearest neighbor. Larger values of k make sense when the topic is represented by multiple vectors, either because the topic is multi-faceted [16] or because the topic's story vectors are never consolidated into a single vector [8].

In those cases, a system might look at several neighbors to estimate the topic of a new story. If it looked at three vectors, for example, it would select the topic that is most common within that small set. Within most of the TDT tasks, even when topics are represented by multiple vectors, k remains one, using a clustering model that is similar in spirit to single link clustering [45].

The reason for this is that topics tend to grow over time and include a wider range of discussion. If the stories in a topic are merged together, the core of the topic is clear, but its edges get lost. Stories on the fringe of the topic would not be considered part of the topic and would incorrectly create new topics. By keeping the topic as a set of distinct story vectors, the range of issues and events discussed within the topic is not blurred.

The downside of this approach is that there are also fringe stories that should not truly be considered part of the topic. For example, they might contain a brief mention of the topic, but primarily discuss a different topic. If not treated carefully, such stories can incorrectly merge unrelated topics together.

Note that $k > 1$ as a nearest neighbor strategy may make the most sense for the tracking task where there are two classes: on topic and off topic. So comparing a story vector to a set of already classified vectors allows a wider range of possibilities. Yang et al. [49] have experimented with a wide range of k-NN strategies to find ones that work best for tracking. Although they were able to improve the effectiveness of their system using variant k-NN methods, the error tradeoffs of the techniques were different enough that it was not clear which would be best. They found that combining results from different techniques addressed much of that concern.

12.4.2 Decision Trees

Another model comparison approach is the use of decision trees. In some ways, a decision tree is really a third type of model. It analyzes some training instances (e.g., stories that are known to be on or off topic) and develops a set of rules for classifying future instances. This approach can be used within the tacking task when there are sufficient number of on-topic training instances. (In theory, tracking can be done with as few as one positive instance of a story, but that is very scant training for building a reliable decision tree.) It has been shown to have acceptable results, comparable to k-NN methods [12], but is not widely adopted, perhaps because it is not as flexible.

The best place for decision trees within TDT may be the segmentation task, where there are numerous training instances (i.e., hand-segmented stories). Finding features

that are indicative of a story boundary (or the absence of a boundary) is possible and achieves good quality results [16].

12.4.3 Model-to-Model

Another style of comparison is direct comparison of statistical language models that represent topics.[‡] The typical way to compare two probability distributions is to use relative entropy, or the Kullback-Leibler divergence, $D(M_1 \parallel M_2)$. The value $D(p \parallel q)$ represents "the average number of bits that are wasted by encoding events from a distribution p with a code based on a not-quite-right distribution q." [27, p.72]. Larger values correspond to less similar distributions. If the distributions are identical, the KL divergence has a value of zero.
Generally in TDT we are interested in a notion of similarity rather than distance, so the number is negated. Therefore larger numbers (closer to zero) indicate more similar distributions. Note that the calculation of the KL divergence:

$$D(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

is not symmetric. However, we generally assume that if story A is on the same topic as story B, then story B is on the same topic as story A. To finesse that problem, we calculate the KL divergence both ways and add them together: $D(M_1 \parallel M_2) + D(M_2 \parallel M_1)$. (And, of course, negate it.)
One of the problems with comparing models is that it is also important that the models be meaningful. Suppose that two models are constructed that are indistinguishable from general newswire text. Those models may be nearly identical by the KL divergence, but it is not useful to know that they are. One approach that has been used to incorporate that notion penalizes the comparison if the models are too much like background news [22]. That is, the initial comparison is replaced by $D(M_1 \parallel M_2) - D(M_1 \parallel \text{news})$. The second value is referred to as "query clarity" [15] since the larger it is, the more the model M_1 diverges from background news, so the less generic it is.[§]

12.5 Miscellaneous Issues

In this section, we talk about several additional issues that can affect how topic models are constructed. TDT allows a modest amount of "look ahead" into the future,

[‡] Arguably, vector comparisons of topic models is also a direct comparison, so this is not strictly specific to language models. However, it is convenient to distinguish between the two for clarity.

[§] Of course, it could still be a terrible model of the topic. However, at least we know it is more specific than general news that could be about anything.

it requires supporting multiple languages (English, Chinese, and more recently, Arabic), and it expects that multiple modalities will appear (viz., newswire and speech recognizer output). Each of those can impact the creation of a topic model.

12.5.1 Deferral

All of the TDT tasks are envisioned as "on-line" tasks that operate on a continuously arriving stream of news. In the limit, that means that a decision about a story is expected before the next story is presented. In fact, TDT provides a moderate amount of look ahead for the tasks.

First, stories are always presented to the system grouped into "files" that correspond to about a half hour of news (newswire stories are grouped together to approximate the same amount of news). A system can do any processing it likes on that entire file before presenting its results for the first story in the file. That means that there is always an effective look ahead of up to thirty minutes (or an equivalent number of stories).

Second, the formal TDT evaluation [19] incorporates a notion of deferral that allows a system to explore the advantage of deferring decisions until several files have passed. Typical values are no deferral (i.e., just the within-file deferral), ten, or 100 additional files.

The advantage of deferring a decision appears for stories that are heavily reported. The extra stories would typically be used by clustering them together and then using the agglomerated super-story to find an appropriate topic. This approach has the advantage that if a new topic appears that is very similar to an existing one, the extra stories might augment the distinction between the new and old topics and decrease the chance of a false alarm. Obviously, the deferral is only useful for topics with multiple stories in the period.

Although several sites have worked with deferral periods, we know of no exhaustive studies to determine their advantage. The TDT pilot study included an infinite deferral period whose purpose was to explore the possibilities of retrospective clustering of news [2].

12.5.2 Multi-modal Issues

The news stories that TDT systems must deal with are either written text (newswire) or read text (audio). Almost all TDT systems used the provided speech recognizer output, though a few worked directly with the audio [41]. (A human-generated transcription of closed caption quality was also provided and used to explore the impact of speech recognition errors.)

On the face of things, it seems that it should be possible to treat both types of text the same. However, speech recognizers make numerous mistakes, inserting, deleting, and even completely transforming words into other (sometimes similar sounding) words. For very clean recordings such as the newscaster reading broadcast news in a studio, the word error rate runs in the 10–20% range [30]. Research in information retrieval has shown little impact on effectiveness from recognition errors as high as

40% [21], but it has not been totally clear the extent to which that would carry over into TDT technology.

One area where the two modes (text and audio) have clear differences are in score normalization. If pairs of stories are compared (in the link detection task) that are from the same mode (i.e., newswire story against newswire story), we get one distribution of scores. If the pairs are instead drawn from audio sources, the distribution is different. And if the stories come from different modes, a *third* distribution appears. This effect means that in order for scores to be comparable no matter the modes of the stories, a system needs to normalize depending on those modes.

The different score distributions might be handled by noting the mean and standard deviation of each distribution. Then, when running the system, the system scores could be shifted according to the appropriate mean and distribution so that all modes have the same mean and roughly the same distribution.

There is some evidence that document expansion smoothing techniques reduce the problem of different distributions, making it simpler to choose single parameters across all modes [22].

12.5.3 Multi-lingual Issues

So far, the discussion of topic modeling has implicitly assumed that all news stories are in English. The TDT research program has strong interest in evaluating the tasks across multiple languages. For TDT 1999 through 2001, sites were required to handle English and Chinese news stories intermixed (though within each file the stories were in the same language). For TDT 2002, sites will be incorporating Arabic as a third language.

The approach that most sites have used is to convert the Chinese stories into English. Since a SYSTRAN translation of every Chinese story is supplied with the corpus, that is the simplest solution and one adopted by many sites. However, other groups are actively researching cross-language information retrieval and related problems, and they did their own information retrieval quality translations of the stories [24, 13]

Similar problems arise in processing cross-language stories as in cross-mode stories. The distributions of scores are different depending on the modes—perhaps even more so in the cross-language case. The SYSTRAN stories, for example, use English in a way that is peculiar to the system's output and quite distinct from human-generated text. That means that SYSTRAN stories are much more likely to be similar to each other than to stories that were originally written in English. The upshot of this is that normalizing the distributions can have an even larger impact than in the cross-mode case [6].

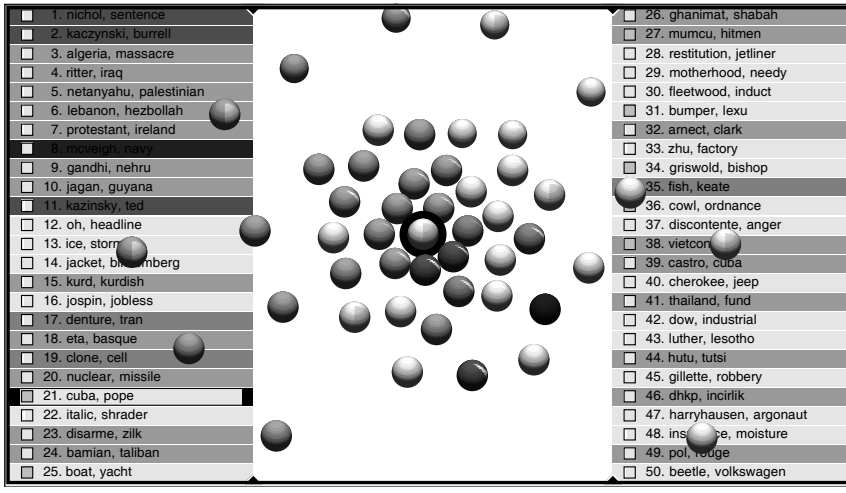

FIGURE 12.2
Screen snapshot of the Lighthouse system that was created to portray TDT topic clusters and their relationships.

12.6 Using TDT Interactively

TDT is viewed as an enabling technology for a range of tasks that want to impose event-based organization on news stories. Ultimately the technology must be incorporated into systems so that people can use it and see whether it is helpful. It has turned out that this type of organization is unfamiliar to people, so is difficult to present. In this section we discuss two ways that TDT technology can be exposed.

12.6.1 Demonstrations

Lighthouse is a prototype system that visually portrays inter-document similarities to help the user find relevant material more quickly [25]. It represents documents as spheres in space and places the spheres such that highly similar documents are nearby in space and the less alike they are, the farther apart they should be. The portrayal of document clustering that is possible in Lighthouse has been shown to allow a statistically significant improvement in effectiveness over IR's classical ranked list.
We created a version of Lighthouse where the portrayed objects were topics and

their relative locations in space indicated similar topics in the news [20]. Figure 12.2 shows a sample screen shot of this prototype. The image would change every time the user asks pending stories to be incorporated into the visualization. (Recall that TDT operates on a stream of news, so stories are constantly arriving while the user is viewing the current state. In this demonstration, the view was held constant until the user asked it to be updated since it was felt that constant uncontrolled shifts in the view would be disconcerting.)

In this demonstration, the user has the ability to search for topics that match a query and to annotate topics with colors so they can more readily be identified at a later viewing. A pie slice is depicted on the top of each sphere that indicates the proportion of the topic made up of stories that appeared since the last time the view was updated. This feature allows highly volatile or entirely new topics to be recognized at a glance. A topic can be selected so that the stories within it can be read.

The Lighthouse-based TDT system was fun and flashy and provided most of the functionality a user would probably like. However, it was awkward to use: people do not understand spheres floating in space and their inter-relationships made no sense. We are currently working on a substantially changed presentation of the same ideas, but based upon the "file folder" metaphor common on computer desktops.

12.6.2 Timelines

Another way of presenting TDT information is using a timeline to show not only what the topics are, but how they occur in time. The interesting part of the problem is finding ways to construct the timeline automatically [43, 44, 42]. The work in this area is not part of TDT directly, but leverages similar ideas.

We start by extracting all names and noun phrases from a collection of news, assuming that names and things are the central components of most of the news. We then scan through those features, considering a day of news at a time. For each day, we use the χ^2 measure to determine whether or not that feature is occurring on that day in an unusual way—typically that means that it occurs much more often on that day than on other days. For example, the word *Oklahoma* occurred in the news much more often shortly after the Oklahoma City bombing in 1995 than it did during the many months of news before that. The χ^2 measure picks that up readily.

Given a set of features that are interesting within a particular time period, we group them together based on whether they co-occur frequently. So *Oklahoma* might group with *McVeigh* but not with other features (e.g., *Simpson* from the O.J. Simpson murder trial) that occurred at about the same time.

Now that a group of features has been found, the collection of stories that talk about that topic is easy to isolate. The timeline can then be constructed as in Figure 12.3. Each topic is depicted as a rectangle. The left-right span of the topic reflects the duration of reporting on the topic. Its area is determined by the total number of stories on a topic: so a flat rectangle has a moderate number of stories over a long period, where a very tall rectangle indicates much reporting in a very short period. The vertical positioning of the rectangle is determined by its "importance" or surprise

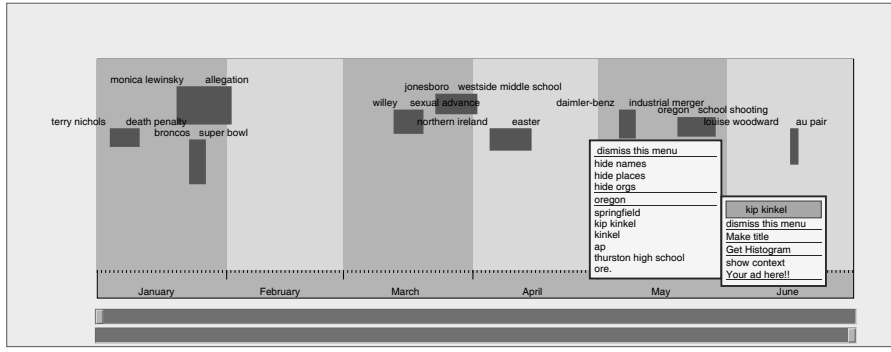

FIGURE 12.3
Overview of January-June 1998. The topic labeled *monica lewinsky allegation* is the highest ranked topic by the χ^2 measure. The pop-up on *oregon school shooting* shows significant named entities for that event. The other pop-up displays a sub-menu for obtaining more information on the name *kip kinkel*.

value based on the χ^2 value of its features. Topics that are very unusual are presented at the top of the graph.

The graph in Figure 12.3 includes the ten most "important" topics in the time period. A nice feature of that importance value is that it can be used to keep 10 topics on the display at any level of granularity: the most important 10 (or any other number) topics in a particular period of time can always be selected.

12.7 Modeling Events

All of the above discussion is about modeling topics within TDT, with a smattering of discussion on presenting the results of the work. Interestingly, almost nothing in the research literature for TDT attempts to model topics as more than a "bag of words" that are weighted appropriately. Topics in the news are related to events. Events are about people. They take place at a particular place and in a given time. With the exception of the use of named entities [17] and the explicit inclusion of time [8] there is almost no recognition that those components of the news topics might be useful. So far, there is virtually no distinction between the technology used for TDT and that used for document retrieval.

That lack of distinction might not be an issue, except that TDT performance is not

adequate for anything other than limited uses: its error rates are still too high. Further, the effectiveness of information retrieval systems appears to have plateaued in the last several years, suggesting that gains in effectiveness are not likely to be had from that direction.¶

It may be possible to incorporate the missing aspects of event-based topics by modeling them explicitly. Eichmann and Srinivasan [17] built different vectors for people, organizations, and so on, compared them separately, and then merged them using a linear combination of the piecewise similarities. This approach was not highly successful, but it clearly captured some of the notions of events. We speculate that the problem is starting this approach from the vector space perspective which gives no theoretical justification or motivation for any of the steps along the way.

Why should names and organizations be separate? What are they trying to capture? Why a linear combination? Perhaps it would be better to look for the who, where, what, and when of news stories: to explicitly model the subject of a topic (or event) or try to identify the location of the happening. Knowing that the system is trying to capture those specific pieces of the event makes it possible to evaluate those items directly rather than evaluating only at the level of topic match.

We have begun some work in this area, hoping to build a rich topic model that captures the various aspects of events. We are motivated by the statistical language modeling approaches and hope that we can extend that well enough to improve effectiveness. Results as of this writing can only be described as promising.

Regardless of whether our work is successful, or whether Eichmann and Srinivasan's approach is the one that works, the important point is that small steps are being made toward modeling events explicitly. TDT tasks are not likely to improve substantially in accuracy as long as broader, more general IR technology is the only approach used.

12.8 Conclusion

We have talked about the topic detection and tracking (TDT) research program and sketched several of the approaches that have been used to address the tasks. We talked about vector space models and statistical language models, the two dominant paradigms in the TDT research literature. We discussed several of the techniques that systems have used to build or enhance those models and listed merits of many of them.

We concluded by talking about the failure of topic models to incorporate the notion of "event" explicitly, relying on technology that is just as useful for the subject-

¶ Information retrieval *research* has not plateaued. The work is being extended into a wide range of new areas and new technologies and ideas appear constantly. After improving 10% a year for several years, ranked document retrieval has not improved substantially since the late 1990s [46].

based topics familiar to IR document retrieval. We believe that TDT researchers have clearly demonstrated the extent to which IR technology can be used to solve TDT problems. However, we also believe that TDT technology must and can be substantially improved, and that the only avenue to that goal is through incorporating information about events into the models directly.

References

[1] J. Allan. Incremental relevance feedback. In *Proceedings of Conference on Information Retrieval Research (SIGIR)*, pages 270–278, Zurich, 1996.

[2] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[3] J. Allan, H. Jin, M. Rajman, C. Wayne, D. Gildea, V. La vrenko, R. Hoberman, and D. Caputo. Topic-based novelty detection: 1999 summer workshop at CLSP, final report. Available at http://www.clsp.jhu.edu/ws99/tdt, 1999.

[4] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of Conference on Information Retrieval Research (SIGIR)*, pages 37–45, 1998.

[5] James Allan. Detection as multi-topic tracking. *Information Retrieval*, Vol. 5, number 2-3, pages 139-157, 2002, Kluwer Academic Press.

[6] James Allan, Victor Lavrenko, David Frey, and Vikas Khandelwal. UMass at TDT 2000. Notebook publication for participants only, November 2001.

[7] James Allan, Victor Lavrenko, and Hubert Jin. Comparing effectiveness in TDT and IR. Technical Report IR-197, University of Massachusetts, Center for Intelligent Information Retrieval, 2000.

[8] James Allan, Victor Lavrenko, and Russell Swan. Explorations within topic tracking and detection. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 197–224. Kluwer Academic Publishers, Boston, 2002.

[9] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 79–85, 1998.

[10] D. Beeferman, A. Berger, and J. Lafferty. Text segmentation using exponential models. In *Proceedings for Empirical Methods in NLP*, 1997.

[11] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using SMART: TREC-3. In *Proceedings of the Text Retrieval Conference (TREC-3)*, pages 69–80. NIST, 1995.

[12] Jaime Carbonell, Yiming Yang, John Lafferty, Ralf D. Brown, Tom Pierce, and Xin Liu. CMU report on TDT-2: Segmentation, detection and tracking. In *Proceedings of the DARPA Broadcast News Workshop*, pages 117–120. Morgan Kauffman Publishers, 1999.

[13] Hsin-Hsi Chen and Lun-Wei Ku. An NLP & IR approach to topic detection. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 243–264. Kluwer Academic Publishers, Boston, 2002.

[14] Christopher Cieri, Stephanie Strassel, David Graff, Nii Martey, Kara Rennert, and Mark Liberman. Corpora for topic detection and tracking. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 33–66. Kluwer Academic Publishers, Boston, 2002.

[15] W. Bruce Croft, Stephen Cronen-Townsend, and Victor Lavrenko. Relevance feedback and personalization: A language modeling perspective. In *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, pages 49–54, 2001.

[16] S. Dharanipragada, M. Franz, J.S. McCarley, T. Ward, and W.-J. Zhu. Segmentation and detection at IBM. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 135–148. Kluwer Academic Publishers, Boston, 2002.

[17] David Eichmann and Padmini Srinivasan. A cluster-based approach to broadcast news. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 149–174. Kluwer Academic Publishers, Boston, 2002.

[18] D. Evans and R. Lefferts. Design and evaluation of the CLARIT TREC-2 system. In *Proceedings of the Text Retrieval Conference (TREC-2)*, pages 137–150. NIST, 1994.

[19] Jonathan G. Fiscus and George R. Doddington. Topic detection and tracking evaluation overview. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 17–31. Kluwer Academic Publishers, Boston, 2002.

[20] David Frey, Rahul Gupta, Vikas Khandelwal, Victor Lavrenko, Anton Leuski, and James Allan. Monitoring the news: a TDT demonstration system. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 351–355, 2001.

[21] J.S. Garofolo, C.G.P. Auzanne, and E.M. Voorhees. The TREC spoken document retrieval track: A success story. In *Proceedings of the Text Retrieval Conference (TREC-8)*, 2000. NIST special publication 500-246.

[22] Victor Lavrenko, James Allan, Ed DeGuzman, Daniel LaFlamme, Veera Pollard, and Stephen Thomas. Relevance models for topic detection and tracking. Technical Report IR-243, University of Massachusetts, Center for Intelligent Information Retrieval, 2002.

[23] Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In *Proceedings of ACM SIGIR Conference on Research in Information Retrieval*, pages 267–275, 2001.

[24] Tim Leek, Richard Schwartz, and Srinivasa Sista. Probabilistic approaches to topic detection and tracking. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 67–83. Kluwer Academic Publishers, Boston, 2002.

[25] Anton Leuski and James Allan. Lighthouse: Showing the way to relevant information. In *Proceedings of the IEEE Symposium on Information Visualization 2000 (InfoVis 2000)*, pages 125–130, 2000.

[26] Anton Leuski and James Allan. Improving realism of topic tracking evaluation. Technical Report IR-248, University of Massachusetts, Center for Intelligent Information Retrieval, 2002.

[27] C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999.

[28] A. Martin, G. Doddington, T. Kamm, and M. Ordowski. The DET curve in assessment of detection task performance. In *EuroSpeech*, pages 1895–1898, 1997.

[29] David Miller, Richard Schwartz, Ralph Weischedel, and Rebecca Stone. Named entity recognition from broadcast news. In *Proceedings of the DARPA Broadcast News Workshop*, 1999.

[30] Long Nguyen, Spyros Matsoukas, Jason Davenport, Jay Billa, Rich Schwartz, and John Makhoul. The 1999 BBN BYBLOS 10xRT broadcast news transcription system. In *Proceedings of the 2000 Speech Transcription Workshop*, 2000.

[31] David D. Palmer, John D. Burger, and Mari Ostendorf. Information extraction from broadcast news speech data. In *Proceedings of the DARPA Broadcast News Workshop*, 1999.

[32] Ron Papka. *On-line New Event Detection, Clustering, and Tracking*. PhD thesis, University of Massachusetts, 1999.

[33] Ron Papka and James Allan. Topic detection and tracking: Event clustering as a basis for first story detection. In W. Bruce Croft, editor, *Advances in Information Retrieval: Recent Research from the CIIR*, chapter 4, pages 96–126. Kluwer Academic Publishers, 2000.

[34] Ron Papka, James Allan, and Victor Lavrenko. UMASS approaches to detection and tracking at TDT2. In *Proceedings of the DARPA Broadcast News Workshop*, pages 111–125, 1999.

[35] J. Ponte and W.B. Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR*, pages 275–281, 1998.

[36] Jay Ponte. *A Language Modeling Approach to Information Retrieval*. PhD thesis, University of Massachusetts, 1998.

[37] Jay Ponte and W. Bruce Croft. Text segmentation by topic. In *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 113–125, 1997.

[38] Stephen Robertson and David A. Hull. The TREC-9 filtering track final report. In *Proceedings of the Text Retrieval Conference (TREC-9)*, pages 25–40, 2001.

[39] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983. Chapter 5, pages 157-198.

[40] J. Michael Schultz and Mark Y. Liberman. Towards a "universal dictionary" for multi-language IR applications. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 225–241. Kluwer Academic Publishers, Boston, 2002.

[41] Andreas Stolcke, Elizabeth Shriberg, Dilek Hakkani-Tur, Gokhan Tur, Ze'ev Rivlin, and Kemal Sonmez. Combining words and speech prosody for automatic topic segmentation. In *Proceedings of the DARPA Broadcast News Workshop*, 1999.

[42] Russell Swan and James Allan. Extracting significant time varying features from text. In *Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM99)*, pages 38–45, 1999.

[43] Russell Swan and James Allan. Automatic generation of overview timelines. In *Proceedings of SIGIR*, pages 49–56, Athens, Greece, 2000. ACM.

[44] Russell Swan and David Jensen. Timelines: Constructing timelines with statistical models of word usage. In *Proceedings of KDD 2000 Conference*, pages 73–80, 2000.

[45] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition, 1979.

[46] Ellen M. Voorhes and Donna Harman. Overview of the eighth text retrieval conference (TREC-8). In *Proceedings of the Text Retrieval Conference (TREC-8)*, pages 1–23, 2000.

[47] Steven Wegmann, Francesco Scattone and Ira Carp, Larry Gillick, Robert Roth, and Jonathan P. Yamron. Dragon systems' 1997 broadcast news transcription system. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[48] Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems (TOIS)*, 18(1), pp.79–112, 2000.

[49] Yiming Yang, Jaime Carbonell, Ralf Brown, John Lafferty, Thomas Pierce, and Thomas Ault. Multi-strategy learning for TDT. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 85–114. Kluwer Academic Publishers, Boston, 2002.