INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Methods for More Efficient, Effective and Robust Speech Recognition

Michael Galler
School of Computer Science
McGill University, Montreal

December 21, 1999

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of the
requirements for the degree of Ph.D.

Abstract

In the last few years the field of computer speech recognition has come into its own as a practical technology. These advances have been due primarily to a steady, step-wise refinement of existing techniques, and the availability of ever more powerful computers on which to implement them. However, many problems remain partially unsolved or entirely open. For example, the current methods continue to fail in cases of small, highly confusable vocabularies.

New acoustic modeling techniques and search methods are proposed in the area of robust, speaker-independent continuous speech recognition. Randomized search techniques are used to refine and improve the topologies and clustered training of allophone hidden Markov models. A new approach for word hypothesization and pronunciation modeling is proposed, based on pseudo-syllabic units. Algorithms are described for transforming a lattice of syllables into words, and learning the phonotactics of syllables automatically in a statistical framework. Next, a multi-grammar method for generating alternative hypotheses is presented, together with the method's experimental evaluation in a telephone-based spelled word recognition system. A blueprint is given for building time and memory-optimized speech decoders. Finally, analysis and recognition techniques are applied to the problem of enhancing the intelligibility of speech produced by alaryngeal speakers.

Abstract

Ces dernières années, l'essor de la recherche en reconnaissance automatique de la parole a permis le développement d'applications concrètes dans ce domaine. Ces avancées sont essentiellement dues à un travail minutieux d'amélioration de techniques existantes, ainsi qu'à l'apparition d'ordinateurs plus puissants sur lesquels ces solutions ont pu être implémentées. Ainsi, plusieurs logiciels de dictée automatique grand vocabulaire connaissent actuellement un véritable succès. Cependant, de nombreux problèmes liés à la reconnaissance restent encore mal résolus.

Cette thèse étudie plus spécifiquement des vocabulaires difficiles où la confusion entre les mots peut être grande, comme les lettres anglaises épellées. À partir de ces lettres, un très grand nombre de noms propres peuvent être composés. Or, une bonne reconnaissance de tels vocabulaires n'est pas correctement réalisée par les systèmes de dictée les plus connus. Plusieurs problèmes liés à la modélisation acoustique et à la reconnaissance de la parole isolée et continue sont considérés ici du point de vue de l'exploration des graphes. Des techniques de recherche aléatoire sont ainsi utilisées afin d'affiner et d'améliorer la topologie et l'apprentissage des modèles de Markov des allophones. Nous proposons également des algorithmes pour transformer des treillis de syllabes en mots, et pour apprendre automatiquement les aspects phonotactiques des syllabes, le tout dans un cadre statistique. De plus, une méthode de recherche dans de multiple grammaires permettant de générer plusieurs hypothèses est présentée. Nous avons évalué expérimentalement cette méthode dans un reconnaisseur basé sur des mots épellés à travers le téléphone. Une méthodologie de construction des moteurs de recherche optimisés en temps et en mémoire est proposée pour des applications de ce type. Dans une étude parallèle, nous avons également appliqué ces techniques d'analyse et de reconnaissance au problème d'amélioration de l'intelligibilité de la parole produite par des locuteurs alaryngiques.

Acknowledgements

This thesis is the outcome of several years of continuous involvement in the field of speech recognition, both in the university and in industry. It comprises a collection of ideas, techniques and applications, most of which have been published elsewhere, in conferences or journals, and worked out in collaboration with a number of excellent colleagues and mentors. Foremost in the latter category are Dr. Renato De Mori and Dr. Jean-Claude Junqua, who have given generously of their time, creativity and patience, without which I could not have finally accomplished this work.

The work on phoneme modeling (chapter 8), pronunciation modeling and syllable phonotactics was performed at McGill University's Center for Intelligent Machines (CIM). At CIM I benefited from sharing software and having discussions with Dr. Charles Snow, Matteo Contolini, Fabio Brugnara, and Giulio Antoniol. Chapter 10 on building an efficient decoder is the outgrowth of a series of discussions with a young colleague, Luca Rigazio, at Speech Technology Laboratory (STL). It began as a tutorial and ended in a collaboration. Many excellent ideas described here are his. The papers and related patents on which I based chapter 11, on robust spelled word recognition over the telephone, were co-authored with Dr. Junqua.

The work on esophageal speech enhancement, in chapter 12 is something of a departure. The problem was introduced to me by the linguist Dr. Hector Javkin, and he provided the data to which we applied techniques of speech recognition and analysis. I found this work particularly rewarding, going as it does beyond the usual goals of improved productivity. It exemplifies the application of computer science for humane purposes. Indeed, what is often overlooked in the whole area of computer speech recognition is that in addition to its many commercial applications, it will serve to significantly improve the lives of the hearing and speech impaired. The chapter is based on a paper and two patents co-authored with the linguists Dr. Javkin, Dr. Nancy Niedzielski, and Jim Reed (all formerly at the University of California - Santa Barbara), and Robert Boman, of Speech Technology Laboratory.

About the software packages and tools I used in my experimental work: I was able to advantage myself of excellent work made available to me in the two speech laboratories where most of this research was performed. The signal acquisition software was in some cases my own; in others I used software written by Philippe Boucher at CIM, and by Dr. Philippe Morin and his colleagues at STL. The MFCC signal analysis program I used was written

at CIM by Dr. Charles Snow. The PLP domain signal analysis software was shared with me by my STL colleagues. One HMM training package I used was shared with CIM by our friends at the Italian research institute IRST. At STL we use HMM training software written in-house by Matteo Contolini, and Dr. Roland Kuhn. All the decoders and search engines used in this thesis research, as well as many of the language building tools were the result of my own development work.

Now to personal matters; I would like to thank the members of my family to whom I owe so much: my father and my late mother, and my wife, Kayoko, whose patient support, tolerance, and encouragement allowed me to pursue and eventually complete this work.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Preface

The investigation described in this thesis took place precisely during the period computer speech recognition moved from the science laboratory to the marketplace. Many of the key innovations and techniques that made this possible were already known 10 years ago. A steady stream of refinements on these methods by speech laboratories around the world, added to the brisk pace of hardware improvements in memory and computational power, has made possible the realization of real-time, large-vocabulary, speaker-independent recognizers. It has also resulted in the widespread deployment of extremely accurate small-vocabulary recognizers in telephone autoattendants. This has made the technology visible and familiar to the general public.

These developments have been a source of pride and excitement for us, the scientists and researchers, and also a keen spur to competition. With the advent of software dictation products such as IBM's *ViaVoice* and Dragon's *Naturally Speaking*, it may seem that the basic research problems have been solved. The truth is that the field of automatic speech recognition (ASR) has emerged from its infancy, with many open problems remaining and many of the current techniques unsatisfactory.

Much remains to be learned about speech and speech recognition, including fundamental issues and practical ones. In 1994 Roger K. Moore observed that "in the end, statistics is just a sound mathematical approach for modeling uncertainty or *ignorance*... [When] speech is fully understood, there may be very little residual uncertainty remaining to be modeled and the stochastic approach will have both served and lost its purpose" [Mak84]. In the meantime our tools for speech recognition remain primarily statistical ones. Where progress has been made is in managing the size of the space we choose to model with statistics. Instead of trying to model the raw signal, we extract features such as the coefficients of the log magnitude spectrum, or the linear prediction model (chapter 5), and try to capture the statistics of this greatly reduced space. The still limited success of recognizers in decoding spontaneous speech (around 50% accurate [BeaufaysEtAl99]) proves that these smaller feature sets do not correctly separate the irrelevant from

the relevant information in the acoustic signal. And yet all the necessary information is there, packed in with a high degree of redundancy.

To illustrate the redundancy of information in the speech signal, consider the bandwidth of the recorded speech signal, and contrast that with the information content, measured in bits per second (bps), of the message encoded in the signal.

If the signal is sampled at 16kHz, in order to capture approximately the frequency range of the human auditory channel, 16,000 two-byte samples of linearly encoded speech are represented per second; at a bit rate of 256,000 bps. For an average speech rate of two English words per second, eight letters per word, the actual information content of the speech signal is 128 bps. The ratio of the signal bit stream to the bit stream of the message means that an enormous amount of redundancy is employed by nature in order to ensure that the message can be decoded, despite numerous variabilities of speech production and transmission [Atal99]. These variabilities are due to the imprecision of vocal production in the individual, individual differences of the vocal tract, differences due to sex and age, differences of dialect and accent, the characteristics of the transmission channel (e.g. microphone, telephone circuit, room echo), and environmental noise, etc.

To the extent that these variabilities are understood, and can be modeled directly, the quantity of ignorance which we model with statistics is diminished. Speech knowledge will eventually allow a better "clustering" in a large acoustic space than the current Gaussian clusters trained on spectrum derived coefficients. This is because the new clusters will be perceptually based, i.e. they will represent units that are perceived as the same phoneme by a human listener. Despite ongoing, and incrementally successful, efforts to diminish the ignorance through improved speech knowledge, we do not yet know enough to properly sift the redundancies inherent to the speech signal.

Thus in 1999, *all* ASR methods are based largely still on statistics. It can be safely said that only moderate qualitative knowledge has been gained about the fundamental nature of the speech signal that serves to reduce the uncertainty. Ignorance reaches from one endpoint of the problem space to the other; from the extraction of the most appropriate spectral (or time-domain) features that feed the speech model, to the domain of natural language understanding (NLU). Without NLU, speech can never be fully decoded in general, and the recognition problem is increasingly coupled to progress in NLU, as applications extend beyond speech-to-text dictation to computer programs that understand and act upon the meanings of natural speech inputs.

1.1 How this thesis is organized.

This thesis is divided into four main parts. The first part consists of this preface and the following two chapters. It presents the context for the ASR problem, describing it in terms of machine learning and pattern recognition.

Part II comprises a survey of the basic tools used for continuous speech recognition. Fundamental algorithms for acoustic analysis and modeling, decoding and search are described.

The chapters of Part III contain original work on problems of acoustic modeling, and the design of a fast search engine for speech decoding. The research described here does not focus on very large vocabularies of common words. The focus instead is on difficult vocabularies, such as spelled letters, with which a very large number of proper nouns can be composed. This is the major application explored in this thesis, a subject for which good solutions are not found in the most popular dictation systems.

The first chapter of Part III describes experiments with acoustic models, their representations and training. The second chapter motivates the use of syllabic modeling and reasoning for search, and enhanced word hypothesization. The next chapter is a short manual on how to optimize the speed and space requirement of a basic speech decoding engine. This search engine is particularly well designed for small vocabularies. The following chapter describes some original algorithms and their resulting improvements to the robustness of a letter-based continuous spelled word recognizer. All four of these chapters are variations on the theme of *search*, particularly in the space of small and confusable vocabularies.

The last chapter of Part III introduces a successful application of speech recognition to the medical issue of esophageal speech enhancement.

The final part of the thesis is a concluding chapter describing the outstanding issues in the field of ASR, and the most current and widely pursued techniques to advance the technology. Also discussed are the weaknesses and strengths of current methods, and a perspective on where the most effort should be spent in order to achieve the next generation of automatic speech and natural-language processing.

Chapter 2

Speech Recognition: A Survey

Automatic Speech Recognition (ASR) is a problem in the transfer of a highly redundant information stream across a noisy interface. The signal originates as a mental representation of a sequence of linguistic symbols, ie. words. The words are translated into sound by complex excitations of the organs of the vocal tract. This signal is received and converted by the auditory mechanisms of the listener's inner ear into a pattern of nerve-cell firings. Within the auditory, perceptual and language centres of the listener's brain, the signal is converted into a sequence of linguistic symbols. The listener then integrates and interprets these symbols as a sentence.

Between ear and mind, or microphone and computer memory, lies an interface across which information is translated from analog to digital, from continuous to discrete, and from physical to symbolic. The challenge is to produce a machine which can reconstitute the original message as quickly and with as little apparent error as a human being.

ASR is more difficult than many problems in pattern recognition because the sound wave encodes the signal in an inherently non-uniform way. Some sources of variability are:

1. Words are made up of *phonemes*, the basic alphabet of sounds constituting a particular language. But different speakers will use a somewhat different sequences of phonemes for a given word, due to a different abstract mental representation of the word.

2. Because of the complex way sounds are articulated, two individuals will not pronounce a given phoneme exactly the same way.

3. Similarly, the same person will not pronounce a given phoneme twice exactly the same way.

4. The duration of a phoneme is variable.

5. Phonemes are distorted by the contextual (or *co-articulation*) effects of neighbouring phonemes.

6. In general, the boundaries between the discrete symbols of interest (the words) are not distinguished with silences or other cues.

In spite of this variability, speech is intelligible. This is because the speech utterance is generally a highly redundant signal, and human beings use information at different levels – acoustic, syntactic, semantic and pragmatic – in order to interpret it.

People have studied the acoustic-phonetic properties of phonemes for many years, and there is now a rich literature in the use of statistical methods and pattern classification techniques for modeling them. The most popular methods include *hidden Markov models* [Rabiner89] and *artificial neural networks* [WeibelEtAl89]. Other approaches include *multi-level acoustic segmentation* and *stochastic segment models* [KimballOstendorf92]. All these techniques have had similar success, and there is reason to suspect they are now nearly at the information-theoretic limit for accurate decoding of speech into sub-word acoustic symbols.

At present more attention is being focused on the study of robust signal processing, acoustic model adaptation, lexical representation, language modeling, and interpretation methods. The goal is to understand better how words relate to articulated sounds, and how context relates to words, so that ASR systems can be designed to work in more general contexts and under more variable conditions than possible with current techniques. Lexical access is a search problem: given a lexicon of words and their symbolic representations, how can the phonetic symbols detected in an acoustic signal best be mapped to a set of word hypotheses. Lexical search must be fast and admissible.

Language modeling is based on a grammar which is in some cases a stochastic one. The language being recognized may be restricted to the rigid syntax of a finite-state grammar, or may be modestly constrained with sequence-based probabilities. In the second case probabilities are associated with the network transitions. In either case the generation of sentences is based on a network of linguistic symbols.

The limits of computer speech recognition algorithms today are due to

1. Limited understanding of the psycho-acoustical factors at work in both the production and the reception of speech sounds.

2. Signal processing algorithms which only guess at the important, information bearing time and spectral-domain features of speech.

3. An acoustic representation that operates at a fixed resolution of discrete intervals.

4. Language models that are not much more than crude filters.

5. Virtually no hard understanding of higher order language processing.

2.1 Historical Review

Computer-based speech recognition can be seen to have evolved through several generations, roughly coinciding with the decades from the 1950s on. Early efforts, which began after the arrival of computers equipped with A/D converters, were based on hardware circuits. An input signal was passed through a bank of analog filters, and the harmonic resonances of the signal were used to spot vowels, or distinguish short words like digits based on their vowels. The early work (1952–Bell Labs, 1956–RCA) showed that the frequency-domain of the speech signal, the spectrum, could be used to tag invariant features of the phonemes for classification. In 1959 at University College, England, Fry and Denes introduced several ideas that would later prove important, including phoneme recognition, the use of a pattern matcher, and a primitive language model based on statistics of phone sequences.

The 1960s saw the introduction of methods for *feature extraction* including zero-crossing analysis, and time normalization of the speech utterance. Vintsyuk, in the Soviet Union, introduced a technique, later popularized as *dynamic time warping* (DTW), which became a generic method for the isolated word recognition.

This 1960s also saw early work in connected word recognition, notably at Carnegie-Mellon University. During this period the first commercial companies were created to market special-purpose hardware for word recognition.

The following decade saw the application or innovation of basic methods that exist in systems of today. These include pattern classification, dynamic programming, frame-synchronous analysis, the distance method for linear predictive coding (LPC) coefficients (Itakura, 1975), network representations, beam search (Baker, 1975), and clustering algorithms for continuous speech (AT&T Bell Labs).

Template matching techniques such as DTW were successfully applied to isolated-word, speaker-dependent applications. About this time, a more general and powerful class of statistical models was recognized by researchers to have the potential to manage continuous speech. The first description of *hidden Markov models* (HMMs) was given in [Baum72]. In [Baker75] their utility in the realm of speech recognition was demonstrated for the first time.

Certain ongoing projects that saw their genesis in the 1970s became benchmark systems for the field. One such system, IBM's Tangora, was a speaker-adaptive isolated word recognition program for dictation of office memos. By 1985 the system was capable of recognizing 5,000 words, and became the basis of a commercial dictation product for personal computers. A comparative overview of ASR in the 1970s which includes work carried out in the non-English speaking world can be found in [Demori79].

In the 1980s two things happened which significantly advanced the state of the art, and disseminated effective tools for the speech recognition problem to a wider community of researchers. The first was the arrival of power-

ful workstations that could perform the demanding numerical computations needed for speech recognition. The second was the popularization of hidden Markov models as a basic methodology [Ferguson80], [Rabiner89].

Other important innovations of this period were methods of integrating phoneme spotting with word recognition for continuous speech, (NEC's two-level dynamic programming, and Bell Labs' *level building* approach), cepstrum-based feature extraction (Shikano), the use of the Mel-scale for transforming the cepstral coefficients [DavisMermelstein80], and the use of neural networks for phoneme recognition eg. (Lippman, 1987) (Weibel, 1989).

In the 1980s great attention was paid to integrating more knowledge sources within HMM-based search engines, e.g. lexical and language models. By this time several outstanding research systems were well-known to be capable of speaker-independent, continuous word recognition for medium-sized vocabularies with good accuracy (> 90% on read speech). Notable systems included CMU's SPHINX, and BBN's BYBLOS, as well as projects of Lincoln Labs, MIT and Bell Labs.

The 1990s saw the culmination of this work leading to the widespread introduction, particularly in the last three years, of commercial systems for speaker-independent continuous speech dictation of very large ($\approx$ 60,000 word) vocabularies, and spoken language understanding in limited domains. ASR technology today, though still imperfect, has acceptable performance for certain applications. Examples include medical and legal report dictation, data entry, and information retrieval by telephone.

ASR applications for the automobile and telephone are now facing the problem of improving performance in spite of low-quality microphones or transducers, limited bandwidth, and environmental and channel noise. Some of the remaining technical challenges include decreasing cost, increasing robustness, capturing speech with far-talking microphones or microphone arrays (giving more freedom to the speaker), and improving models so that confusable vocabularies and natural speech are properly recognized.

References: [Demori98], [Rabiner89], [Lee89], [O'Shaughnessy87].

Chapter 3

Machine Learning and Pattern Recognition

The field of Artificial Intelligence (AI) has been defined as "the study of ideas which enable computers to do the things that make people seem intelligent" [Winston77]. Two endowments in particular mark human activity as intelligent: the ability to solve difficult problems, and the ability to form abstract representations, or *models* of the world, models which serve as the basis for further problem-solving and new abstractions.

These two activities, model-building and problem-solving, are intimately related. The formation of a new model is a kind of problem to be solved, called the *learning problem.* The models we build are useful mostly to the extent they afford new solutions to problems in some domain, solutions that would otherwise be harder to attain. In trying to automate learning and problem-solving, we find that different kinds of models emerge naturally from different domains.

For example, there are *rule-based models*: symbolic models based on rules and facts. Expert systems are tools for building rule-based models of the world. These models are natural ways to simulate expertise in medical diagnostics, or tax law [Tanimoto87], expertise that depends on basic knowledge, and accumulated experience of how facts relate to one another.

The rule-based model embodied in an expert system has two main components: a database of *facts*, and a set of *rules of inference.* The rules allow the expert system to consider combinations of facts, and from them, infer new facts. The system takes the facts at hand and attempts to generate appropriate answers to the questions being asked. This is accomplished by a separate component to the system called an *inference engine.*

The inference engine makes problem-solving in a rule-based model automatic, but the learning is not. The model is built through a laborious preliminary stage of data acquisition, and often involves interviewing human domain-experts. (Note that a distinction is being drawn here between *learning*, the gathering of an initial set of rules and facts, and *problem-solving*, the subsequent automatic generation of new facts. In an expert system this

distinction can be arbitrary; the latter is also sometimes described as a stage of learning).

In contrast with rule-based models, there are *functional models*: numerical models based on functional relations of real-world variables. Scientists construct functional models in seeking to understand physical systems. Many human abilities such as vision and speech involve the processing of vast streams of physical data. These data are hard to describe adequately with symbols or facts, but can be sampled and quantified. Problems such as these lend themselves to functional models.

In many cases there are automatic means for training functional models, called algorithms for *machine learning*. Let $M = \{X, Y\}$ be a measurement of a set of variables, where Y is a vector of the variables we wish to consider dependent on X, the remaining variables. Training a functional model means taking a series of such *observations*, or measurements M_i, and finding the parameters P of the model $f(X) \equiv f(P, X)$ which produce the best *fit* of the model-predicted values $(X_i, f(X_i))$ to the actual measured (X_i, Y_i). Outputs can be smoothly continuous functions, as in the case of regression models, or discrete elements of a finite set, as in pattern classification.

Algorithms for fitting problems involve optimizing some measure of closeness of fit; thus the training of functional models relies on optimization methods.

In summary, AI deals in large part with building models and using them to solve problems automatically. Rule-based models are usually trained by hand, but functional models can be "machine-learned" using optimization algorithms. Problem-solving with rule-based models centres on the generation of inferences. Problem-solving with functional models varies with the kinds of models, but in general involves computations on numerical inputs.

The preceding was an attempt to illustrate the strong relation between machine learning, specifically the fitting of parametric models, and computer methods of optimization. The next sections survey the subjects of fitting and optimization in order to provide context for the following work. Chapter 6 will discuss in detail the parametric model to be used, hidden Markov models.

3.1 The Fitting Problem

3.1.1 Regression Methods

One approach to the fitting problem is to assume the data are perturbations of the values of a "true" model, and the differences between the true values and the measured values, or the *measurement errors*, are drawn from a known probability distribution. Models assuming particular distributions are called *parametric* models in statistical literature, because they require estimation of the parameters of the distributions.

Stating the problem in statistical terms allows one to solve it by tech-

niques of *maximum likelihood estimation (MLE)*; finding the set of model parameters with the highest likelihood given the data.

The classical techniques of *linear regression* apply MLE to the simplest fitting problem: given a set of n points $p_i \equiv (x_i, y_i)$ in the real-number plane, find the straight line $y = A+Bx$ which best fits these points. The coefficients A and B are estimated from p_i using the chi-square merit function:

$$\mathcal{X}^2(A,B) = \sum_{i=1}^{n} \left(\frac{y_i - A - Bx_i}{\sigma_i} \right)^2$$

If the x_i are known exactly, the y_i are reported with normally distributed errors, and σ_i is the standard deviation for the measurement error in y_i, then minimizing $\mathcal{X}^2$ gives A and B with the highest likelihood. The minimum can be computed directly.

The usefulness of this estimate depends on a negligible *fixed component*, or error due to model inaccuracy, in the statistical error. In most applications the linear model is only an approximation. In many cases a linear model is inadequate even as an approximation.

By modeling the data with a linear combination of m arbitrary functions $X_1(x), ..., X_m(x)$, the regression approach can be extended to non-linearly-related data . In the *general linear least squares* method, the parameters for the model are a set of m coefficients A_i, and the merit function becomes

$$\mathcal{X}^2 = \sum_{i=1}^{n} \left(\frac{y_i - \sum_{k=1}^{m} A_i X_k(x_i)}{\sigma_i} \right)^2$$

The vector A which minimizes $\mathcal{X}^2$ is solved for by methods of LU or *Singular Value* decomposition [PressEtAl88].

3.1.2 Pattern Classification

In ordinary regression the dependent variables are continuous. *Pattern Classification* is the fitting problem in which the values of a dependent variable range over a finite set of discrete elements.

Consider the independent variables $x_1, ..., x_n$ as *features*, and the values taken on by the dependent variable as the *classes*. This kind of problem involves learning a classifier function which separates feature space $\Re^n$ into disjoint *decision regions* corresponding to the classes. The regions are described in terms of their boundaries, which divide *clusters* of point sets in feature space, each cluster corresponding to a class.

If the clusters are *linearly separable*, the decision boundaries can be modeled as a set of hyperplanes, $H_i = W_{i1}x_1 + ... + W_{in}x_n$, in the feature space. If, on the other hand, the cluster distributions are multimodal, the decision surfaces are non-linear and require non-linear models.

Discriminant Analysis

Bayesian discriminant analysis is the application of parametric MLE techniques to pattern classification. The goal is to derive a classifier with the smallest likelihood of error. This is any function which, for class C_i and feature vector $\mathbf{x}$, maximizes the *a posteriori* probability $P(C_i|\mathbf{x})$. One such function is

$$\max_i(\log p(\mathbf{x}|\mathbf{C_i}) + \log \mathbf{P}(\mathbf{C_i})).$$

$P(C_i)$, the *a priori* probability of class C_i, is estimated simply from cluster size. The difficulty is in properly estimating $p(\mathbf{x}|\mathbf{C_i})$, the conditional probability densities of the feature variables. Bayes classifiers generally assume uni- or multi-variate normal densities.

Linear discriminant analysis assumes that the clusters all share the same feature-covariance matrix. In this case the classes are hyperplane-separable, and the the decision-bounding hyperplanes are given by

$$H_i = -\frac{1}{2}(\mathbf{x} - \mu_i)^t \sum^{-1} (\mathbf{x} - \mu_i) + \log \mathbf{P}(\mathbf{C_i})$$

where $\sum$ is the covariance matrix and μ_i is the mean feature vector of cluster i. Maximizing H_i yields the classes with minimal likelihood of error [DudaHart73].

Dropping the assumption that each class shares the same covariance matrix leads to a slightly more complex, quadratic model. The decision boundaries become *hyperquadratics*, and can separate clusters with more complex distributions. The models' appropriateness still depends on how realistic are the assumptions made for the feature densities.

Hidden Markov Models

Hidden Markov models (HMMs) are the basic classification method used in this thesis, and are described in detail in Chapter 6. The following is a brief introduction, placing them in context of the family of pattern matching techniques.

A Markov chain is a stochastic process which models systems of events in sequence. It consists of a set of *states*, and the *transitions* between them. At any given discrete time t_i the system is in one of a finite (or countable) set of states; in the next discrete time t_{i+1} the system makes a transition to a new state. For each state the process has a set of probabilities associated with transition to other states (figure 3.1.) These probabilities must sum to 1.

The entire system is described by a *state transition matrix*. A matrix corresponding to the Markov chain of figure 3.1 with real transition probabilities is given in figure 3.2.

In the techniques of previous sections each classification is statistically independent of previous classifications. A Markov chain is useful for modeling

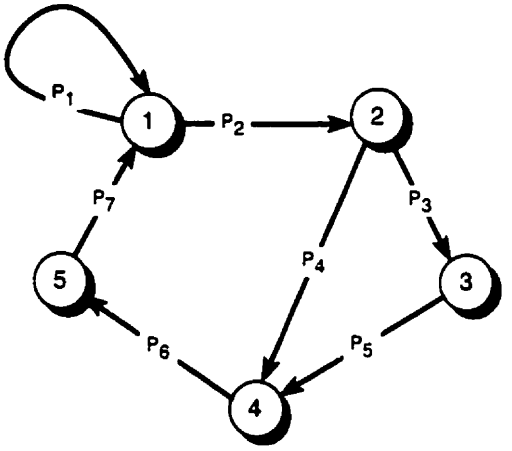

Figure 3.1: A Markov chain.

	1	2	3	4	5
1	0.7	0.3	0.0	0.0	0.0
2	0.0	0.0	0.5	0.5	0.0
3	0.0	0.0	0.0	1.0	0.0
4	0.0	0.0	0.0	0.0	1.0
5	1.0	0.0	0.0	0.0	0.0

Figure 3.2: The state transition matrix.

serial processes, in which prior events affect the likelihood of subsequent events. For example, the likelihood of a system being in state X at time t_j given that the system started in state Y at time t_0 can be computed with a set of operations on the state transition matrix, (equivalent to multiplying the appropriate transition probabilities.) Similarly, calculations on the transition matrix can determine the likelihood of observing a particular sequence of state-transitions $S_i -> S_j -> S_k$ This process could be called an *observable* Markov model, since the outputs of the process are states corresponding to observable events in the system being modeled.

A richer model is derived by embedding a second stochastic model in the Markov source, such that the observable events of the system no longer correspond to the states, but rather are generated as a probabilistic function of the states. Since the states are not observable, the models are called *hidden Markov models* (HMMs). The state of the system at time t can be inferred through the stochastic process that produces the observable events. The latter processes are modeled with probability distribution functions, in practice, Gaussians, whose parameters depend on the state.

HMMs are trained for pattern classification by modeling sequences of feature vectors as observable events, and using them to estimate both the output distributions and the state-transition probabilities of the Markov chain. Here, too, the objective is MLE: find the parameters of the HMM with the maximum likelihood given the data.

The number of states in an HMM and its transition-structure are *not* functional parameters; they are selected empirically or on the basis of some intuition about the system being modeled. In speech the most commonly employed topology is a feed-forward structure (proposed by Bakis, 1976) with a beginning state on the left and a terminal state on the right.

HMMs satisfy two assumptions about the stochastic processes being modeled. The *state* stochastic process assumes that when the model is in a given state at time t, the history before time t has no influence on future events – the so-called "first-order Markov hypothesis." The *observation* stochastic process assumes that neither past states nor past observations affect the present observation if the last two states are specified – the "output independence hypothesis." Because hidden Markov models produce sequences of real-valued observations, they are particularly well-suited to model temporally evolving physical processes like speech.

Principal Component Analysis

In addition to the parametric models described above, there are many non-parametric techniques for pattern classification, techniques which do not assume particular probability distributions for the features.

Principal Component Analysis (PCA) is a parameter transformation technique which can be applied to certain pattern classification problems. PCA is a method of reducing n-dimensional vectors made up of statistically cor-

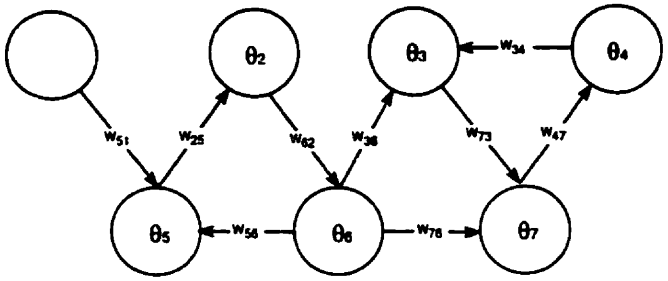

Figure 3.3: A perceptron.

related features into smaller r-dimensional vectors of uncorrelated features. The *Singular Value Decomposition* (SVD) theorem tells us that the rows of a matrix of m feature vectors can be expressed as linear combinations of r orthogonal vectors, (*principal components* in statistics,) where $r \leq n$ [Stewart73].

The principle components are an alternative set of orthogonal coordinate axes, and in this new coordinate space two different variables have zero covariance. Projecting the data along the principal component with the largest variance is equivalent to finding the direction of greatest variance in the feature space. When this direction coincides with the least overlap between data clusters, PCA can perform pattern recognition. For example, PCA can classify acoustic signal as speech or as noise because speech samples have singular vectors with large variance relative to samples of noise [BakamidisEtAl90]. PCA differs from discriminant analysis which attempts to project explicitly along the direction of maximum discrimination between classes.

Perceptrons

Perceptrons (also called *Neural Networks* or *Connectionist systems*) are a family of powerful, non-parametric pattern recognition tools which have become the subject of intensive investigation in recent years. Perceptrons are distributed networks of simple processors, called *artificial neurons* in analogy with human brain cells (see figure 3.3.)

The neurons communicate via *connections* between them. Each unit has a state that is computed as a function of the inputs received along connections from other units. The combined activities of the units operating in parallel lead to complex behaviors.

The kinds of functions governing *activations*, or state-changes, in a neuron vary with the connectionist model. One typical such function is a weighted linear sum:

$$x_j = -\theta_j + \sum_i y_i w_{ji} \tag{3.1}$$

where y_i is the state of unit i, w_{ji} is the weight on the connection from unit

1st Input	*2nd Input*	*Output*
0	0	0
0	1	1
1	0	1
1	1	0

Table 3.1: The XOR operator.

i to unit$j$, and θ_j is a *threshold* term which acts as a bias on the j'th unit's activation. In more complex networks, the new state of unit j will usually be a non-linear function of the linear input x_j.

Neural Networks have been applied to such diverse areas as associative memories, data compression, pattern classification, and speech recognition (for examples in the area of speech, see [RobinsonFallside90] and [BengioEtAl90]).

The units of a perceptron are assigned to specific groups or *layers*, including an *input* and an *output* layer. For pattern recognition problems, the units in the input unit layer have their states set according to the pattern of feature variables. The neurons then update their activation-states accordingly, and the output-unit states are interpreted as a class representation. The connection weights are determined such that when all cells in the network have computed their new states as a function of their individual inputs, the activations of the output cells represent the correct classification of the input pattern. Finding a set of weights to enable the perceptron to classify effectively is another variant of the fitting problem.

There are many training algorithms available for fitting the connections weights to the data. The *perceptron convergence theorem* showed that the weights could be automatically learned [Block62] by iteratively computing, for each input pattern, a change in w_{ji} according to the learning rule

$$\Delta w_{ji} = \epsilon(t_j - y_j)x_i$$

where t_j is the desired state of neuron j (found in the training data), y_j is the actual state, and ϵ is a constant representing the learning rate.

Since this algorithm requires knowledge of the correct activations for all the units in order to fit the connection weights to the data, it can only be used to train two-layer perceptrons, ie. networks containing only input and output cells. Unfortunately, as Minsky and Papert proved in [MinskyPapert69], certain functions such as topological connectedness and parity are *not* computable by two-layer networks. In fact, no set of weights can enable a two-layer network to model a non-linearly-separable function. A well-known example of such a function is the exclusive-or (XOR) operator (table 3.1).

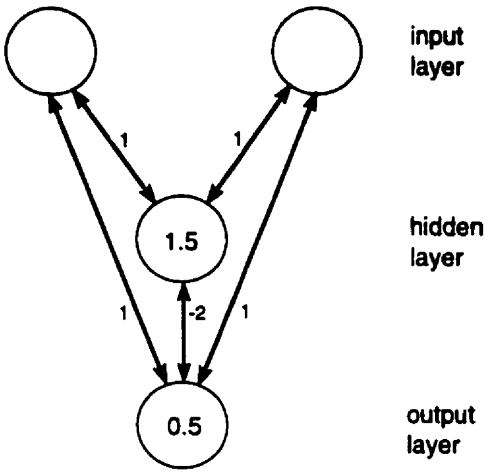

Figure 3.4: A multi-layer neural network to compute XOR.

Multi-layer Neural Networks

Nonlinear problems can be solved by adding a layer of *hidden* units which allow the perceptron to build internal representations, ie. to model higher-order relations in the feature data. Figure 3.4 gives a simple example of a network to solve the XOR problem, (figure adapted from [RumelhartMcClelland86]). In the figure, the linear activation rule of equation 3.1 computes the XOR function of the states of the input units.

The problem then becomes how do we find the weights for hidden units, ie. how do we solve the fitting problem for multi-layer perceptrons. One solution is the *error back-propagation* algorithm. In this method an error estimate E is computed as the sum of squares of the output activation errors:

$$E = \frac{1}{2}\sum_{j,c}(y_{j,c} - t_{j,c})^2$$

where c is an index of the training patterns. This error is differentiated with respect to the weights on input connections, beginning with the output layer, and working backwards. As the error propagates back through the network, it is used as a negative factor of proportionality in the weight modification rule Δw_{ji}. The weights are thus modified in the direction of minimal activation error. This is easily recognized as a form of *gradient descent*, which will be discussed in the section on optimization methods.

Another way to derive weights in a multi-layer perceptrons is the *Boltzmann Machine* algorithm [AckleyEtAl85]. This is a stochastic method that modifies the connection weights in the direction of convergence between the conditional probability distribution exhibited by the output units of the perceptron and the desired conditional probability distribution. The advantage

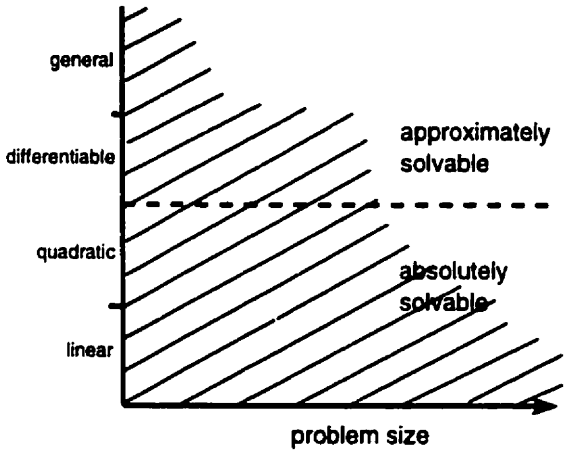

Figure 3.5: Tractability of optimization problems.

of this non-parametric algorithm is that it computes the *a posteriori* probabilities of the outputs. Also, it is able to model the feature distributions without making any restrictive assumption on the types of distributions. However, the algorithm runs quite slowly.

3.2 Optimization Methods

As described above, machine learning problems are solved with optimization techniques. The latter are as varied as the former. In general, the optimization problem is stated:

Maximize (minimize) objective cost function:

$$f(x_1, ..., x_n)$$

Subject to m constraint functions:

$$g_i(x_1, ..., x_n) \leq 0$$

Where:

$$i = 0, ..., m.$$

Optimization problems can be divided into two groups; linear/quadratic problems, for which large cases can be solved absolutely; and general problems, for which the solution space must be exhaustively searched to find an optimum (and consequently can only be solved approximately.) The shaded area of figure 3.5 shows how the size of tractable optimization problems varies with the class of problems (graph adapted from [Fletcher69].) The linear problem, in which both the objective function and the constraint functions are linear combinations of the input vector X, is solved with the *simplex*

algorithm, a subject addressed in detail by any elementary textbook of linear programming. Methods for the general category of problems include *hill-climbing* techniques and *randomized search* methods. Differentiable problems are solved with hill-climbing algorithms.

3.2.1 Hill Climbing

Nothing in general is known about the solution of the general problem in the present of constraints. Hill-climbing methods for optimizing general problems without constraints fall into two basic categories: *gradient* techniques, and *direct search* methods.

Hill-climbing is a way of incrementally improving the solution to f by exploring the solution surface and climbing (or descending) toward a maximal (minimal) solution. This approach suffers from the fact that in any sufficiently interesting problem the n-dimensional solution space has complex contours with many local extrema. Once a local solution is found, hill-climbing either terminates or must begin again from a new initial solution; worse, for continuous functions there is no general way to identify a globally optimal solution. (Of course, finite discrete problems can be solved optimally through exhaustive search, but only for cases sufficiently small to avoid combinatorial explosion.)

Gradient methods are for problems in which the gradient vector

$$(\frac{\partial f}{x_1}, ..., \frac{\partial f}{x_n})$$

can be computed. Obviously this restricts the method to continuous, differentiable cost functions. Since the partial derivatives provide a measure of improvement in f as the solution X is perturbed in a particular direction, computing the gradient allows one to modify the solution at each step in the direction of fastest change toward a local extremum.

Gradient techniques include *conjugate gradient* and *quasi-Newton* methods. Most weight-correction rules for perceptron training, including the *back-propagation* algorithm, are gradient-descent methods. So is the Baum-Welch algorithm for MLE fitting of Markov model parameters (chapter 6.) Given the advantage of having a computable gradient, it can still be difficult to ensure these methods make efficient progress along the solution surface, even toward a merely local optimum. For example, in the method of *steepest descent,* each successive point is found by minimizing along the line from the previous point in the direction of the local downhill gradient. Steepest descent in a long, narrow, three-dimensional valley can lead to an inefficient search, if the initial point does not happen to be on the plane normal to the short axis of the valley (figure 3.6, (adapted from [PressEtAl88]).

When no gradient is available, hill-climbing can only be guided by repeated direct evaluations of the cost function itself. These *direct methods* involve sectioning or bracketing an area of the solution space in which a local

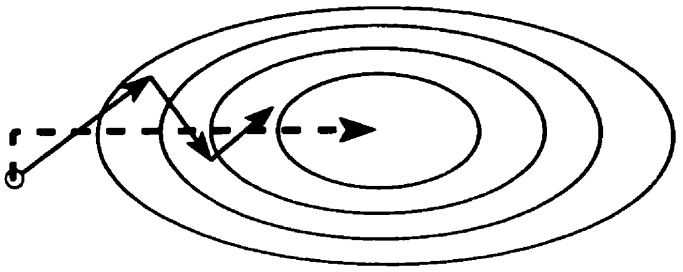

Figure 3.6: Inefficiency of steepest descent. Dotted line shows desired minimization.

minimum can be found, and iteratively continuing search within that section. Examples include *golden section search* and the *downhill simplex method.*

3.2.2 Randomized search

In contrast with Hill-Climbing, randomized search does not set out to thoroughly explore the vicinity of a local extremum of the cost function, but rather employs a stochastic solution generator to visit points all over the solution surface. One advantage of this approach is that it is easily applied to almost *any* optimization problem, finite or infinite, continuous or discrete. Clearly the chance that a randomly generated set of m solutions $\{(x_{11}, ..., x_{1n}), ..., (x_{m1}, ..., x_{mn})\}$ will contain the global or even a local extremum of f is negligible for non-trivial cases. However, there is a higher probability of yielding a solution S^* such that

$$|f(S^*) - f(S_{opt})| < m$$

where m is some acceptable margin for error in the solution. In the case of finite, combinatorial optimization, if there are z possible solutions and e random solutions are generated with uniform probability $\frac{1}{z}$, the probability that one of these will lie in the top l solutions is given by $P(\frac{l}{z}|e) = 1-(1-\frac{l}{z})^e$.

The pure random search above is sometimes called the *Monte Carlo* method. *Simulating annealing* is a method of randomized optimization which is effectively a hybrid between Monte Carlo search and Hill-Climbing, beginning like the former and settling into the latter near the end of the search. The intriguing thing about simulated annealing is that it is the *only* known general optimization method that can provide global solutions.

3.2.3 Heuristic Search

Many discrete optimization problems can be formulated such that the solution space is a tree or graph to be expanded one node at a time. In this formulation, nodes are steps along the path to a solution.

Heuristic search is an algorithm which uses domain (problem-specific) knowledge to direct the order in which nodes are visited. A useful heuristic is one that directs the search to the optimal solution before the search becomes exhaustive.

The A^* algorithm is a heuristic search procedure that employs a an evaluation function $e^*(n) = g^*(n) + h^*(n)$, with $h^*(n) \leq h(n)$, where g^* estimates the minimum cost of a solution passing from the start node to node n, and $h^*(n)$ estimates the minimum cost of a solution passing from n to the solution. $h(n)$ is the true cost of the path from n to the solution. It is easy to show that if the descendants of visited nodes are queued by order of increasing $e^*(n)$, and nodes are expanded from the head of the queue, that the first completed solution path will represent an optimal solution. Graph search problems such as the *traveling salesman problem* naturally lend themselves to A* heuristic search.

Heuristic search is obviously an improvement over exhaustive search. However, if the lower bound estimate h^* is very much lower than $h(n)$, or the estimate is expensive to compute, the search time will be on the order of the size of the search space. The important point to note in this context is that discrete functions can be optimized using both randomized and knowledge-directed search.

3.3 Hand-Tuning of Machine Learning Models

The parameters of machine-learning models are fitted to training data using appropriate methods of optimization. For example, the statistical distributions of hidden Markov Models, and the connection weights of perceptrons are both estimated by gradient descent algorithms.

In both learning models, however, there are structural parameters which are set by hand, either empirically or because of some understanding of the specific learning problem being modeled. The designer of a neural network has to decide the number of neurons assigned to each layer, and the number and type of connections. Suppose the network is intended to map a set of 5 features to one of 3 classes: small, medium, and large. Since the solution to the learning problem will compute a transformation from the input 5-coordinate space to three values along a single dimension, one successful network topology might be the 4-layer perceptron in figure 3.7. Unit T represents the projection of the input space onto a single dimension.

Markov models have *topology*: the number of states and the transition paths between them. These presumably reflect the hidden states and visible events of the process being modeled. Yet in many practical cases, including ASR, the underlying discrete states of the system are unknown, if they exist at all.

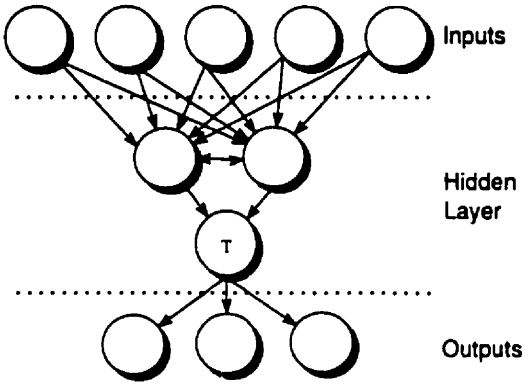

Figure 3.7: Adapting perceptron architecture to problems.

Part II

Fundamentals of Speech Processing

Chapter 4

Speech Production and Acoustic-Phonetics

4.1 Phonemes and Allophonic Variability

The human vocal tract can produce an infinite variety of sounds. However, every language can be characterized by a basic set of abstract linguistic units called *phonemes*. The phonemes are the smallest set of sounds adequate to represent the phonology of the language. Typically there are 20-50 phonemes in a language, and they constitute an alphabet of sounds to uniquely describe words in the language. Table 4.1, borrowed from [LeeEtAl90], lists the phonemes of the English language with examples.

The physical sound produced when a phoneme is pronounced is called a *phone*. Phonemes are discrete units; however, the vocal tract is not a discrete system. The same phoneme will be produced in a slightly different way by each speaker, and by the same speaker with each articulation. Thus, the phoneme is only an exemplar corresponding to an infinitely large class of phones.

Another source of variability is context. The continuously spoken speech signal is not a concatenated sequence of discrete phones with a 1-1 correspondence to phonemes. Because of the smoothly changing nature of the articulatory process, a phone is affected by its proximity to the phones pronounced both before and after. This effect is called *coarticulation*. The term *allophone* describes a class of phones corresponding to a particular variant of a phoneme.

4.2 Speech Production by Phoneme Group

There are two sources of speech sounds: vocal cord vibration, and *frication*, or turbulent noise produced by forcing air past a constriction in the vocal tract. Phoneticians classify speech sounds according to *manner* of articulation, *place* of articulation, and *voicing*. Manner of articulation is concerned with airflow:

Phoneme	*Example*	*Phoneme*	*Example*	*Phoneme*	*Example*
iy	beat	l	led	t	tot
ih	bit	r	red	k	kick
eh	bet	y	yet	z	zoo
ae	bat	w	wet	v	very
ix	roses	er	bird	f	fief
ax	the	en	mutton	th	thief
ah	but	m	mom	s	sis
uw	boot	n	non	sh	shoe
uh	book	ng	sing	hh	hay
oy	boy	d	dad	zh	measure
aw	bough	g	gag	dx	butter
ow	boat	p	pop	el	bottle
ao	bought	ch	church	sil	-
aa	cot	jh	judge	cl	-
ey	bait	dh	they	vcl	-
ay	bite	b	bob	epi	-

Table 4.1: English phonemes.

the paths it takes, and the degree to which it is impeded by vocal tract constrictions. Place of articulation is the point of narrowest constriction. Voicing means there is a quasi-periodic vibration of the vocal cords as a phone is produced.

Phonemes can be divided into phonetic groups and subgroups which share acoustic characteristics. These relations are summarized in tables 4.2 and 4.3.

The two main groupings are vowels and consonants. Vowels are voiced phonemes produced by vibrating the open vocal tract. Consonants are produced with a relatively narrow constriction at one of eight regions in the vocal tract; thus they are closely associated with place of articulation.

For example, English consonants comprise *labials*, produced by the lips; *dental* sounds; *alveolar* sounds, produced by the tongue near the uvula (figure 4.1 - 6); *palatovelar* sounds, produced by the tongue near the palate; and *glottal* sounds, produced by closed or constricted vocal folds.

The consonants are usually grouped according to their manner of articulation. English consonants come in five such groups: *plosives, fricatives, nasals, liquids,* and *affricates.* These groups are acoustically dissimilar. Within the consonant groups, phonemes are distinguished by voicing and place of articulation.

Plosives, (also called *stops,*) come in two groups: voiced (*b, d, g,*) and unvoiced (*p, t, k.*) Plosive sounds consist of a building up of pressure behind a total constriction, followed by a release. The point of closure for a *p* or *b* is the lips; for a *t* or *d*, the tongue pressed to the hard palate; and for a *k* or

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

Figure 4.1: The vocal tract. (from [DemoriEtAl90])

Class	*Manner-of-Articulation Subclass*	*Phoneme*
vowel	short vowel	ih
		eh
		ae
		ix
		ax
		ah
		uh
		ao
		aa
		ao
		er
	diphthong	iy
		ey
		ay
		oy
		aw
		ow

Table 4.2: Vowel classification.

g, the tongue pressed to the soft palate.

Fricatives also come in voiced and unvoiced groupings: *v*, *th*, *z*, and *zh* are voiced; and *f*, *dh*, *s*, and *sh* are unvoiced. Voiced fricatives are produced by vocal cord vibration and excitation at the glottis (figure 4.1 - 14.) The air flow becomes turbulent at the point of constriction; for example, *z* is articulated at the upper incisors. The unvoiced fricatives are produced by a steady flow of air from the lungs, past the open glottis, to the vocal tract constriction.

The nasals *m*, *n*, *ng*, and *en*, are always adjacent to a vowel. They are caused by a closing of the oral cavity and opening of the *velum* (figure 4.1 - 5) during the articulation of the preceding vowel, effectively *nasalizing* it. The vocal tract constricts at some point; the place of articulation distinguishes the nasal consonant.

Liquids (also called *semi-vowels* or *glides*) are produced by a constriction in the vocal tract smaller than that of vowels, but still large enough to avoid frication. This group of consonants, which include *w*, *y*, *r*, *l*, and *el*, are distinguished by a slower rate of articulatory movement.

The affricates *ts*, *ch*, and *jh*, are plosive/fricative pairs: *t-s*, *t-sh*, and *d-zh* respectively. The affricates are often modeled as single phonemes because of the relatively short duration of the trailing fricative.

The vowels are voiced, unless whispered, and are louder than consonants. English contains about 15 different vowels. The characteristics of the different vowels are shaped by the location of the tongue, position of the jaw, and the degree of lip rounding. The vowels can be grouped in space according to tongue position: back, front, and central. They are also grouped as ordinary

Class	*Manner-of-Articulation Subclass*	*Phoneme*
consonant	liquid	l
		el
		r
		y
		w
		hh
	nasal	en
		m
		n
		ng
	fricative	ch
		jh
		dh
		z
		zh
		v
		f
		th
		s
		sh
	plosive	b
		d
		g
		p
		t
		k
		dx

Table 4.3: Consonant classification.

vowels, eg. *ih, ah;* and *diphthongs* such as *iy* and *ay*. The diphthongs can be thought of as two different ordinary vowels spoken in sequence. Phonetically, they are realized as a changing vowel sound in which the tongue and lips move between two vowel positions.

Chapter 5

Analysis of the Speech Signal

5.1 The Acoustic Analyser Module

The previous text mentions "features" or "parameters" used to train the machine learning model, without giving much indication what these are. This chapter is about the specific feature data we use to train statistical models of human speech.

Figure 5.1 illustrates the flow of information along the speech channel. The intended utterance of the speaker becomes an acoustic signal $s(t)$, generated by the articulatory organs of the vocal tract. This signal is received and converted by the auditory mechanisms of the listener's inner ear into a pattern of nerve-cell firings. Within the auditory perceptual and language centres of the listener's brain, this received signal $r(t)$ is converted into a sequence of linguistic symbols $\mathbf{S}$. These symbols might be words, or subword units, (or non-verbal locutions). The listener then integrates and interprets these symbols as a sentence.

The labels along the top of the figure describe the human modules of speech perception. The bottom labels describe the corresponding machine modules of an automatic speech recognition system. An acoustic signal is transformed into an electrical waveform by a microphone, and then into a sequence of numerical samples by an analog-to-digital converter. The sampled waveform must then be fed to an *acoustic analyser* in order that a pattern, $\overline{P}$, of acoustic features sufficient for its interpretation be extracted. These features are passed to the final module, the *linguistic symbol generator*, which matches acoustic patterns to linguistic symbols.

Chapter 6, on hidden Markov models, discusses our choice for a symbol generator. This chapter is about the acoustic analyser, or *feature extractor*. In both human and automatic speech recognition, the acoustic features used to discriminate among linguistic symbols are correlates of articulatory features. In Chapter 4's brief description of speech production, a range of relevant articulatory features were introduced. This chapter describes the processing of the acoustic signal into various compressed representations which

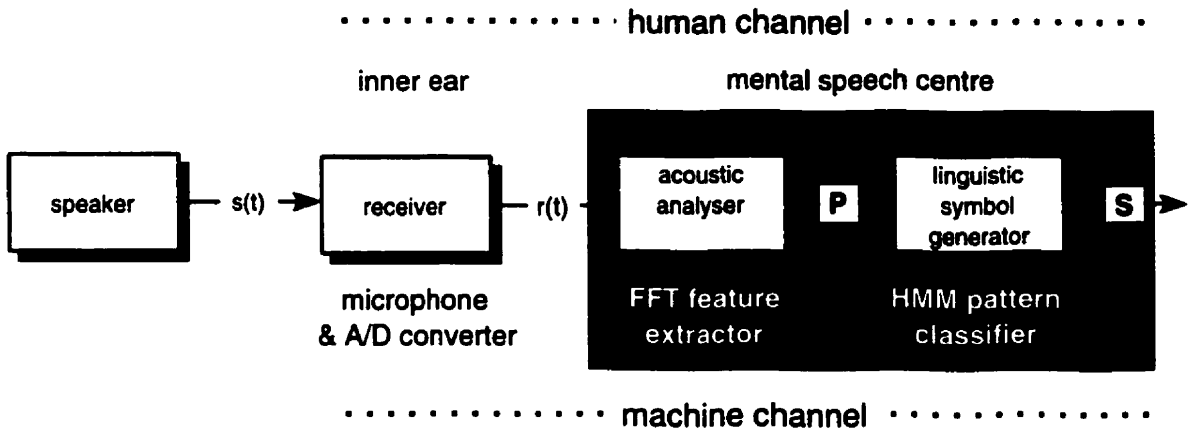

Figure 5.1: Transformation of information along the speech channel.

capture articulatory features, and are used to train speech models. The kinds of analysis include measures of signal *energy*, observed in the time domain, *linear predictive coding* (LPC) based analysis, *Mel-filtered cepstral coefficients* (MFCC), and *perceptual linear predictive* (PLP) analysis. The latter three are computed in the frequency domain.

5.2 Signals and Spectra

Signals have many properties including amplitude, periodicity, duration, and *energy*. Energy is a measure of the signal intensity over an interval of time I:

$$E_x = \int_I x^2(t)dt. \tag{5.1}$$

The *time waveform* of an acoustic signal is a plot of the signal amplitude versus time. Although the time waveform contains all signal data, the information is encoded in a form not easy to interpret. Two speech samples that sound the same to a human being may have time waveforms that look quite different.

Spectral analysis uses the Fourier transform to represent signals in terms of frequencies. The production and perception of speech sounds are more effectively described in frequency terms, because phonetic features are more apparent in the frequency domain than in the time domain.

The *spectrogram* or *spectral display* converts the two-dimensional waveform into a three-dimensional pattern: amplitude versus frequency versus time. Time is the horizontal axis, frequency is the vertical, and amplitude is denoted by the darkness of the display. Figure 5.2 contains the time-waveform and spectrogram of a speech sample. The waveform shows the intensity, periodicity and duration of speech segments. The spectrogram

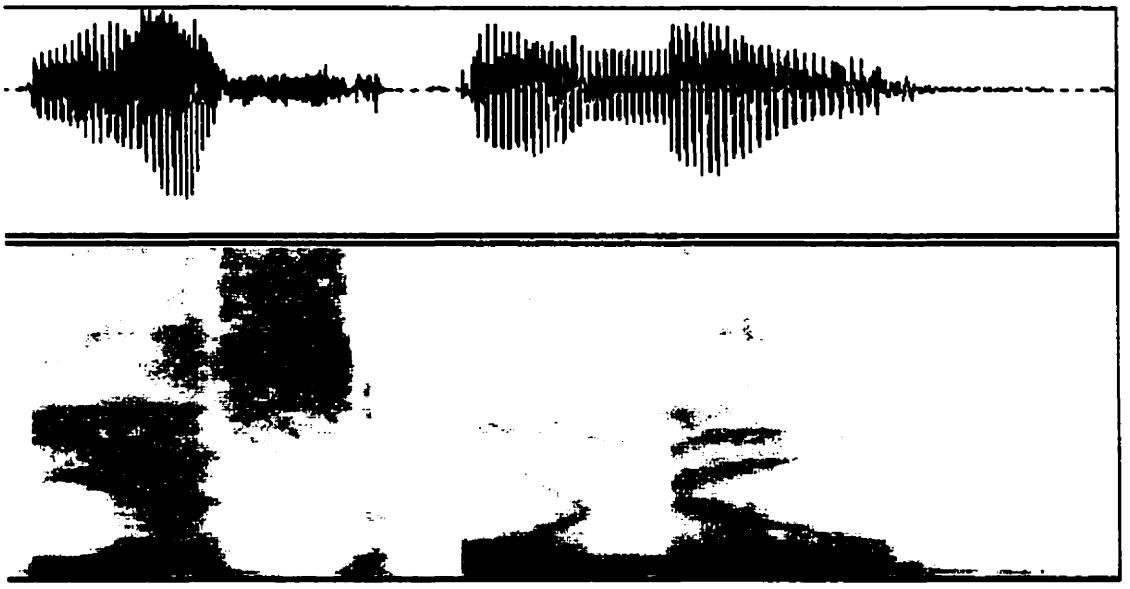

Figure 5.2: Waveform and spectrogram of an acoustic sample: "yes..or no?".

shows the distribution of speech energy as a function of frequency.

Voiced sounds are produced when the vocal folds open and close rapidly, generating a quasi-periodic excitation of the vocal tract. The rate of vibration is called the *fundamental frequency*. The tract, acting as a filter, increases the energy at certain sound frequencies while attenuating others. These spectral energy peaks, called *formants*, show as darkened frequency bands across a spectral display. In the figure, the formants of the vowels *e* and *o* can be seen as 4 dark horizontal bands across a shifting range of frequencies. The distribution and movement (in time) of the formants and energy troughs, or *resonances* and *anti-resonances*, appear to be the primary acoustic cues to phonemes for the human ear [O'Shaughnessy87]. They also work well for the purposes of automatic speech recognition.

5.3 Acoustic Properties of Phonemes

What follows is a brief survey of the principle acoustic properties of the different phoneme groups. These properties appear to be the acoustic cues by which phones are recognized by the human ear. They may be described as a *necessary but not sufficient* set of cues for recognition: phonetic variability due to speaker, dialect, context, and random variation necessitates that higher-level knowledge be employed to map the signal to a phonemic string.

Vowels are characterized by substantial energy in frequencies up to 3500Hz. Energy is concentrated in spectral lines at multiples of the fundamental frequency. Vowels can be distinguished from one another by the locations of

their first three formants. In general, when the tongue moves forward in the articulation of a vowel, the second formant rises. As the tongue moves higher, the the first formant decreases. Lip-rounding lowers the formants. The variances of formant frequency distributions about the means for each vowel are speaker-independent [DemoriEtAl90].

Nasals are manifested as a sharp change in the intensity and spectral features of a vowel, resulting from the entry of the air flow into the nasal cavity. They are marked as well by a low-frequency spectral peak at around 300Hz.

The liquids are distinguished from adjacent vowels by a shift in formants. If formants are labelled in order as $F_1, F_2, \ldots, F_i$, phoneme *l* will be marked by a lower F_1 and F_2 and a higher F_3 than the adjacent vowel. Liquid formant transitions are slower than those of other consonants.

The waveforms of fricatives and plosives are very different from the periodic, or *sonorant* phonemes above. They are aperiodic, much less intense, and contain most of the energy at high frequencies. Voiced fricatives often have simultaneous noise and periodic sound with some low-frequency energy at the onset of frication. Unvoiced fricatives are shorter in duration.

Plosives are acoustically characterized as a prolonged silence followed by an abrupt increase in amplitude at the moment of release. They are *transient* rather than steady-state phenomena; in this they differ from other phonemes. Release is accompanied by a frication *burst*. The interval between release and the onset of voicing for the following vowel is longer for unvoiced (30 to 60 ms) than for voiced (10 to 30 ms) plosives.

As mentioned before, the vocal tract is not a discrete mechanism. The various organs of articulation move smoothly and slowly between positions for different phonemes. They often do not reach the target position due to the contextual effect of neighboring phones. This coarticulation effect means that allophones have different spectra from the target phonemes, which further complicates the recognition task.

5.4 Acoustic Feature Extraction

The speech signal can be considered a non-stationary stochastic process. Thus, its spectral analysis must take into account variability in both time and frequency. As discussed earlier, the signal is produced by articulatory organs moving from one position to another with intrinsic mechanical time constraints. Therefore, it is possible to define a *stationarity interval* within which the signal can be considered time-invariant. During such intervals, standard spectral analysis methods (see following sections) can be applied. The main goal of the feature extraction step is the computation of a sequence of feature vectors that provide a compact representation of the relevant information for the input signal [Demori98].

5.4.1 Signal Preprocessing

Prior to spectral analysis, there are steps of preprocessing which improve the effectiveness of the feature extraction. Preprocessing may be aimed at noise reduction (one such technique is called *spectral subtraction*), or at enhancement of formant visibility in the power spectrum. A characteristic of the power spectrum is that higher-frequency formants have lower energy. *Preemphasis* is a compensatory technique, realized by applying a fixed first-order FIR filter with the z-transfer function $H(z) = 1 - az^{-1}$, where a is the preemphasis parameter (usually 0.95).

5.4.2 Spectral Analysis

True periodic signals are called *sinusoids*. Speech sounds are analyzed in terms of their sinusoidal components via Fourier series. A quasi-periodic signal can be expressed as a linear combination of weighted sinusoids:

$$x_p(t) = \sum_{k=-\infty}^{\infty} c_k \exp(j2\pi kt/T), \tag{5.2}$$

where T is the signal period, and c_k is a Fourier series of coefficients:

$$c_k = \int_{t=T_0}^{T+T_0} x_p(t) \exp(-j2\pi kt/T)dt, \tag{5.3}$$

[j is the imaginary; T_0 is any constant. $\exp(-j2\pi\theta) = \cos(2\pi\theta) + j\sin(2\pi\theta)$.]

As well, *aperiodic* signals can be modeled as sums of weighted sinusoids:

$$x_a(t) = \int_{f=-\infty}^{\infty} X(f) \exp(j2\pi ft)df, \tag{5.4}$$

where $X(f)$ is the *Fourier transform* of $x_a(t)$. The complex frequency function $X(f)$ determines the frequency content of the signal $x_a(t)$. $|X(f)|$ is called the *spectrum*. $X(f)$ is defined

$$X(f) = \int_{t=-\infty}^{\infty} x_a(t) \exp(-j2\pi ft)dt. \tag{5.5}$$

On a computer, the signal $x_a(t)$ is represented as a finite, discrete time-sequence of digital samples, $x(n) = x_a(nT)$, where T is the period of the sampling rate. This leads to a discrete analog for equation 5.5, the N-point *windowed Discrete Fourier Transform* (DFT):

$$\tilde{X}(f) = T \sum_{n=0}^{N-1} x_a(nT) \exp(-j2\pi fnT). \tag{5.6}$$

where N is the size of an analysis segment, or *window*. Clearly, the sampling rate and window size will affect the accuracy of approximation $\tilde{X}(f)$. If N is

infinite, the approximation approaches the Fourier transform as the sampling period T becomes infinitely small.

The *Nyquist sampling theorem* states that the sampling rate need only be twice the highest frequency contained in the signal. In other words, if $X(f) = 0$ for $|f| \geq B$, then then the sampling rate must satisfy

$$\frac{1}{T} \geq 2B.$$

(B is called the signal *bandwidth*). The evaluation of $\tilde{X}(f)$ requires another discrete sampling, this time of the real variable f. Because $\tilde{X}(f)$ is a periodic function, only values from 0 to the period $1/T$ need be sampled. It turns out that setting the frequency sampling interval to $1/(TN)$ allows for a fast computation of the transform. This means that the number of frequency samples computed is equal to the window size N. Computation of the DFT is performed by means of a well-known *divide and conquer* algorithm called the Fast Fourier Transform (FFT).

N is thus a crucial variable in spectral analysis of the speech signal. Small values for N, meaning short windows and DFTs using few points, give poor frequency resolution. On the other hand, they give good *time* resolution. A window of samples is examined for evidence of some "current" aspect of the speech process. It must be short enough so that the acoustic properties of interest do not vary significantly within the window. While a longer window will allow spectral features to be evaluated more accurately, those features will be averaged over a longer evolution of the signal, and rapid spectral changes that are key for recognition may be smoothed away. Most speech analysis uses a fixed window duration in the range of 10-25 ms [O'Shaughnessy87]. To avoid undesirable edge effects, the window's samples are multiplied by a rounded window, with tapered edges, such as the Hamming window. This has the effect of de-emphasizing the importance of the samples near the window's edges on the spectral computation.

Other limitations of estimation methods based on the FFT include

- The position of the analysis window falls randomly within the *pitch period* of voiced sounds. This can cause fluctuations in the spectral estimate, especially for short windows.
- The sample spectrum is a biased estimator of the power spectral density due to the finiteness of window length.

Although spectrographic analysis suffers these drawbacks, it is widely used for speech analysis because the time-frequency patterns it produces have proved useful [Demori98].

5.4.3 Auditory-Based Analysis

Auditory or *ear-model* analysis attempts to model the perceptual processes of the inner ear through a frequency selective linear model implemented with a

bank of filters. These filters are spaced along the frequency axis based on the idea of *critical bands*, i.e. ranges of sound frequency roughly corresponding the tuning curves of auditory neurons. A critical band is a range for which experiments have shown auditory perception will abruptly change as a sound stimulus is modified to have frequency components beyond the band. The most commonly used critical band scales are the *Mel scale* and the *Bark scale*. These scales space filters linearly up to about 1 kHz, and use a logarithmic spacing above 1 kHz.

5.4.4 Linear Predictive Coding Analysis

The basic idea behind the linear predictive coding (LPC) model for speech is that a given sample of speech s_i can be approximated as a linear combination of the preceding n samples, as follows:

$$s_i = \sum_{j=1}^{n} c_j s_{i-j} + G x_i, \tag{5.7}$$

where x_i is an *excitation term* that compensates for the error of the approximation, and G is the gain of the excitation. The coefficients $c_1, ..., c_n$ are assumed to remain constant over the frame of analysis.

Formulation 5.7 leads to a speech model which is both computationally and analytically tractable, and has been widely put to use for many years in speech coding, synthesis and recognition. The LPC model is useful because it provides a good model of the speech signal, particularly during the near steady-state regions of voiced phonemes like vowels. The all-pole model of LPC is a good approximation of the vocal tract "spectral envelope" in these regions, and it permits an analytical separation of the source and vocal tract models, as follows.

By applying the *z transform* to relation 5.7 we derive

$$S(z) = \sum_{i=1}^{n} c_i z^{-i} S(z) + G X(z), \tag{5.8}$$

and a transfer function, relating input and output:

$$H(z) = \frac{S(z)}{GX(z)} = \frac{1}{1 - \sum_{i=1}^{n} c_i z^{-i}}. \tag{5.9}$$

Equation 5.9 has a straightforward interpretation. An excitation source, x, scaled by the gain G, serves as input to the all-pole system $H(z)$ which, acting as a digital filter, produces the speech signal s. This model is used for speech synthesis by feeding either a periodic train of pulses (for voiced segments) or a random noise sequence (for unvoiced sounds) to the filter. The coefficients c_i of the filter model the characteristics of the vocal tract.

The prediction error $e(i)$ is the difference between the true signal and the linear combination of preceding speech samples:

$$e(i) = s_i - \sum_{j=1}^{n} c_j s_{i-j}. \tag{5.10}$$

LPC analysis is performed on successive frames of speech, usually 5-20 msec. long. The basic problem is to determine the set of predictor coefficients that minimizes the mean-squared prediction error over a given speech frame.

For the purpose of speech recognition, the LPC predictor coefficients do not themselves constitute a useful feature set. They are usually transformed, for example, by the *autocorrelation method*, into equivalent representations that are effective. One such set of features is called the *reflection coefficients* [RabinerJuang93]. Another useful set are the LPC-derived cepstral coefficients. The experiments in this thesis are based on cepstral coefficients derived from the logarithm of the signal magnitude spectrum (section 5.4.6), or from perceptual linear prediction analysis, described next. These have been shown in many studies to be superior to LPC for speech recognition.

5.4.5 Perceptual Linear Predictive Analysis

The all-pole model of the speech spectrum estimated by LPC analysis can be viewed as a means of obtaining the smoothed spectral envelope of the speech signal. In [Hermansky90] it is argued that the disadvantage of this model is that the all-pole model approximates the short-term power spectrum equally well at all frequency bands. This violates important known properties of auditory perception:

- Hearing is most sensitive in the middle frequency range of the audible spectrum.
- The spectral resolution of human hearing begins to attenuate above 800Hz.

LP analysis has other problems. The estimated LP pole frequency is often shifted in the direction of the nearest harmonic peak. The inconsistency of estimated LP poles limits the use of LP formant extraction. This is particularly true for female voices, in which formants are often observed to merge [HermanskyEtAl85].

Perceptual Linear Predictive (PLP) analysis modifies the spectrum to mimic characteristics of the human auditory system. This modified spectrum is then approximated by an all-pole model (the linear prediction polynomial) using an autocorrelation method, as in standard LPC analysis. The method consists of the following steps:

1. The speech signal is blocked into 10 ms. frames and Hamming-windowed, using a 20 ms. window.

2. A 256 point FFT is performed for each window, and the power spectrum is computed.

3. The spectrum is warped along its frequency axis using the Bark (critical band) scale. This involves convolution of the spectrum with the simulated critical band masking curve. The result is then sampled in 1-Bark intervals.

4. The sampled spectra are preemphasized with a simulated *equal-loudness curve*, to approximate the disparate sensitivity of human hearing at different frequencies.

5. As an approximation to the power law of hearing, a cubic root amplitude compression is computed:

$$\psi(\omega) = \omega^{0.33}.$$

   This simulates the non-linear relation between the intensity of sound and its perceived loudness.

6. Finally, the spectrum is approximated by an all-pole model using the autocorrelation method of LPC analysis. 8 LPC coefficients are derived.

In the experiments of chapters 10 and 11, the resulting *autoregressive* coefficients are further transformed to produce cepstral coefficients of the all-pole model. This signal analysis is called *PLP-Cepstra*. The derivatives of the PLP-Cepstral coefficients, calculated using 8-sample linear regression, are added to the feature set, as well as the signal energy and its derivative, forming 18-dimension feature vectors.

Another transformation employs time-filtering of the cepstral trajectories to obtain a representation that is particularly robust over the telephone, and other situations where the training and test environments vary. This feature set is called RASTA-PLP [HermanskyEtAl91].

5.4.6 Mel Filtered Cepstral Analysis

The features we employed for the purposes of training some of our HMM speech models were based not directly on the spectrum, but rather on the inverse transform of the logarithm of the speech spectrum $|X(f)|$, called the real *cepstrum*. In terms of the FFT computation, the cepstrum is defined as

$$c(n) = \frac{1}{N} \sum_{f=0}^{N-1} \log |X(f)| \exp(j2\pi fn/N) \tag{5.11}$$

where

$$n = 0, 1, \ldots, N-1.$$

The cepstral coefficients are the coefficients of the Fourier transform representation of the log magnitude spectrum. These have been shown to be a better feature set for speech recognition than the LPC coefficients. The Mel filters are a set of logarithmically spaced, triangular filters designed to bias the feature set toward perceptually important *critical bands.*

More specifically, we use the first 12 *Mel-filtered cepstral coefficients* (MFCC). These are given by

$$\log|X(f)| = 2\sum_{m=1}^{20} C_m cos(fmt) + C_0. \tag{5.12}$$

The following spectral analysis was performed for extraction of features from speech signals in our research. The speech data was digitally sampled at 16kHz, and pre-emphasized with a factor of 0.95. Every 5 ms a 256-point FFT computation was performed (window duration was 20 ms.) The 12 Mel coefficients were calculated; these formed the first part of the feature vector for a speech segment. The spectral information was supplemented by additional information about *rate of change* of spectral features, as represented by the difference cepstrum:

$$\Delta C_i = \frac{\sum_{k=-2}^{2} kC_m(t+k)}{\sum_{k=-2}^{2} k^2}. \tag{5.13}$$

All together, 24 Mel-based cepstral coefficients were extracted from each window of the speech signal.

5.4.7 Waveform Analysis

Time waveform analysis is used to supplement the cepstral analysis of the previous section. (Indeed, it is simpler and more intuitive to examine the untransformed signal for features relevant to discrimination of speech segments. Much early work involved deriving measures directly from the time waveform, e.g. zero-crossing rates.)

The frequency based features are typically supplemented with a measure of energy and its derivatives, as given by (equation 5.1) and the approximate energy rate-of-change (ΔE) below:

$$\frac{\Delta E(t)}{\Delta t} = R[E(t-\Delta t), E(t+\Delta t)] \tag{5.14}$$

R is a linear regression of m (in our case 9) successive samples. These two measures added to 24 Mel coefficients, make up the complete observation vector. Thus, the acoustic analyzer created a 26-entry observation vector to represent the features of each speech segment. This constitutes the acoustic analysis used in the training and testing of speech models for the experiments to be described in chapters 8 and 12.

Chapter 6

Training and Decoding with Acoustic Models

A first order Markov chain is a stochastic process which generates sequences of discrete symbols. It consists of a set of states, and the transitions between them. The system makes transitions from one state to the next in the manner of a non-deterministic finite state automation, but the state transitions are governed by statistics instead of rules. For each state the process has a set of probabilities associated with transition to other states.

A Markov chain is useful for modeling serial processes, in which prior events affect the likelihood of subsequent events. For example, the likelihood of a system being in state x at time t_i given that the system started in state y at time t_0 can be computed with a set of operations on the state transition matrix (in effect multiplying the initial and subsequent transition probabilities.) Other simple calculations can determine the likelihood of observing a particular sequence of transitions.

The Markov chain could be called an *observable* Markov model, since the outputs are state symbols corresponding to observable events in the system being modeled. A richer, second-order Markov model is derived by embedding another statistical model in the Markov chain, such that the observable events of the system do not directly correspond to the states, but are generated instead as a probabilistic function of the states. Since the states are not observable, the second-order models are called *hidden Markov models* (HMMs). The state of an HMM system at (non-initial) time t is not known in general, but can be estimated statistically from the observed chain of events. This doubly-embedded statistic structure renders HMMs capable of modeling non-linear relations between the features and the feature-space classes.

HMMs are trained for automatic speech recognition by casting acoustic feature vectors as observable events, and using them to estimate both the output distributions and the state-transition probabilities of the Markov source. Because HMMs implicitly assume that events near in time are statistically *dependent*, they are well suited for modeling the acoustic-phonetic

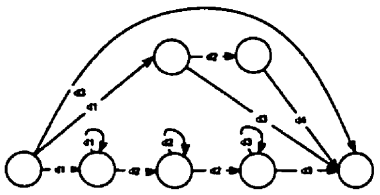

Figure 6.1: Hidden Markov model for a phoneme

patterns involved in ASR. HMMs are also useful because they have a convenient framework for dealing with duration variability. Figure 6.1 is a typical HMM topology for modeling phonemes.

Each transition has an associated probability, and an output distribution function for generating observation vectors. A transition from one state to another causes the production of an observation vector. The HMM's feed-forward structure models the stages of evolution of the speech unit. The skip transitions between non-adjacent states model the durational variability of the phone.

The number of parameters in the overall speech model depends on the number of distinct statistical distributions. Since the accuracy with which the model parameters are trained depends on the amount of training data which is usually limited, it is useful in practice to reduce the dimensionality of the fitting problem by sharing distributions between related transitions.

The outgoing transition probabilities for a given state S must sum to 1. The observation vectors may be composed of discrete or continuous variables. The distribution $d(X)$ associated with a transition t must satisfy

$$\sum_X d(X) = 1, if X discrete;$$

$$\int d(X)dX = 1, if X continuous.$$

A discrete distribution function may take the form

$$d(\underline{X}) = p(X|d), X \in \{0, 1, ..., K-1\}.$$

$\underline{X}$ is a discrete scalar which takes on one of K different values. A more robust discrete model employs multiple codebooks to reduce quantization error:

$$d(\underline{X}) = \prod_{c=1}^{N} p_c(x_c|d),$$

where N is the number of codebooks, x_c is the cth component of X, and $x_c \in \{0, 1, ..., K_c\}$ where K_c is the size of the cth codebook.

Continuous distributions HMMs use N-dimensional multivariate Gaussians:

$$d(\underline{X}) = \frac{1}{|\sum|^{\frac{1}{2}}(2\pi)^{\frac{N}{2}}} e^{-\frac{1}{2}(\underline{X}-\mu)^T \sum^{-1}(\underline{X}-\mu)}$$

where N is the length of vector $\underline{X}$, μ is the mean vector of distribution d, and $\sum$ is its covariance matrix.

Unfortunately, acoustic feature distributions are not unimodal and can't be accurately modeled by Gaussians. Better approximations are possible by use of finite *mixtures* of Gaussian densities [NeyNoll88]. A mixture distribution is a weighted sum of K distributions:

$$P_{mix}(\underline{X}) = \sum_{k=1}^{K} w_k P_k(\underline{X}),$$

where $\sum_{k=1}^{K} w_k = 1$.

Mixture distributions are easily implemented by having several parallel transitions between two states, where each transition is assigned one Gaussian probability distribution. There are also hybrid models called *semi-continuous* HMMs [HuaJac89], which combine discrete probabilities with continuous densities, in an effort to combine the efficiency of the former with the accuracy of the latter.

6.1 Basic HMM Computations

Given a HMM parameter set H, a model m, and a length-r observation vector $y_r = y_1, \cdots, y_r$, the quantity evaluated most often is $P(y_r|m)$, the probability of observations y_r being generated by m. $P(y_r|m)$ can be computed as

$$P(y_r|m) = \sum_{path \in m} P(y_r, path|m) = \sum_{path \in m} P(path|m)P(y_r|path)$$
$$= \sum_{path \in m} \left(\prod_{i=1}^{L_{path}} q_{t_{path,i}} \right) \left(\prod_{j=1}^{r} b_{t_{path,j}}(y_j) \right), \quad (6.1)$$

where L_{path} is the length of the transition sequence $path$, $t_{path,j}$ is the jth transition in $path$, q_t is the probability of transition t, and b_t is the distribution belonging to transition t. Equation 6.1 states that the probability of the model generating y_r is obtained by summing the conditional probability of y_r over every possible path in the model multiplied by the *a priori* probability of that path. (We assume only paths long enough to generate the observation sequence are considered).

The computation of Equation 6.1 is exponential in r. Fortunately, however, there is an efficient recursive procedure to estimate the desired probabilities. To illustrate HMM procedures,we define the following quantities:

$$\alpha_l(i) = P(y_1, \cdots, y_l|m, S_l = i), \quad (6.2)$$
$$\beta_l(i) = P(y_{l+1}, \cdots, y_r|m, S_l = i), \quad (6.3)$$

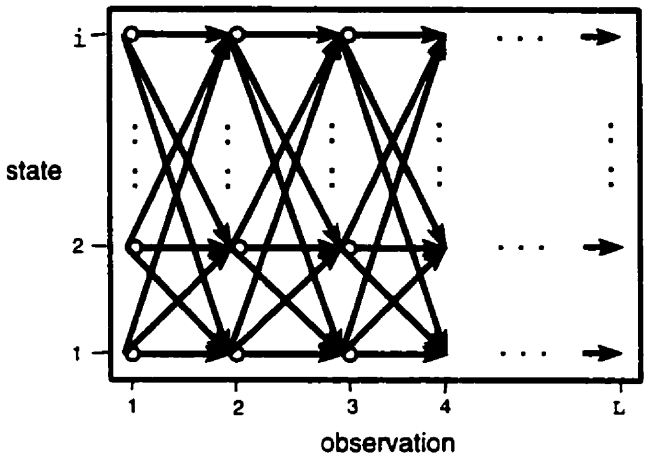

Figure 6.2: A trellis structure for alpha/beta computations.

where S_l is the state producing the observation y_l. $\alpha_l(i)$ is the probability that the model m generates the observations $y_r = y_l, \cdots, y_l$ using a path ending in state i. $\beta_l(i)$ is the probability that the model m generates the observations $y_r = y_{l+1}, \cdots, y_r$ using a path beginning in state i. Thus

$$P(y_r|m) = \alpha_r(S_r) = \beta_0(0). \tag{6.4}$$

The computation of $\alpha_l(i)$ and $\beta_l(i)$ involves an implied data structure called a *trellis* (figure 6.2.) Column l of the trellis corresponds to time l and observation l. Row i corresponds to the ith state in the HMM. $\alpha_l(i)$ and $\beta_l(i)$ are computed recursively , column by column, $\alpha_l(i)$ starting in column 0, and $\beta_l(i)$ starting in column r.

Some transitions of an HMM may have no distribution associated with them, because they model changes of state which produce no observation. These are called *empty* transitions. $\alpha_l(i)$ and $\beta_l(i)$ must be computed in two parts: a sum over empty transitions and a sum over full ones.

For $\alpha_l(i)$, each entry in the trellis is computed, in increasing order of column numbers and state numbers, as follows:

a) Initialization:

$$\alpha_0(i) = \begin{cases} 1, & \text{if } i = 0 \\ 0, & \text{otherwise} \end{cases} \tag{6.5}$$

b) Recursion:

$$\alpha_l(i) = \sum_{t|full, r_t = i} \alpha_{l-1}(l_t) q_t b_t(y_l) + \sum_{t|empty, r_t = i, l_t < i} \alpha_l(l_t) q_t, \tag{6.6}$$

where l_t and r_t are the origin and destination states, respectively, of transition t.

For $\beta_l(i)$, each entry in the trellis is computed, in decreasing order of column numbers and state numbers, as follows:

a) Initialization:

$$\beta_r(i) = \begin{cases} 1, & \text{if } i = S_r \\ 0, & \text{otherwise} \end{cases} \tag{6.7}$$

b) Recursion:

$$\beta_l(i) = \sum_{t|full, l_t=i} \beta_{l+1}(r_t) q_t b_t(y_{l+1}) + \sum_{t|empty, l_t=i, r_t>i} \beta_l(r_t) q_t. \tag{6.8}$$

Both recursions 6.6 and 6.8 are linear in sequence length r. Define $P_l(t, y|m)$ as the probability that observation sequence y was generated by model m, using a path in which transition t was taken at time l. A transition at time l is a transition that reaches a node in the lth trellis column. $P_l(t, y|m)$ may be efficiently computed as

$$P_l(t, y|m) = \begin{cases} \alpha_l(l_t) q_t \beta_l(r_t), & \text{if } t \text{ empty} \\ \alpha_{l-1}(l_t) q_t b_t(y_l) \beta_l(r_t), & \text{if } t \text{ full} \end{cases} \tag{6.9}$$

This quantity is central to most HMM computations.

6.2 Training Algorithm: Maximum Likelihood Estimation

Determining the HMM probabilities is a problem in maximum likelihood estimation (MLE): finding the HMM parameters with the maximum likelihood given the statistics of the training corpus.

Maximum Likelihood Estimation (MLE) by the *Baum-Welch* re-estimation method [Lip82] is the most commonly used training procedure for estimating the parameters of hidden Markov models. Other methods have been used as well, including *maximum mutual information* (MMI) estimation [Bahl86] [Chow90], and *minimum discrimination information* (MDI) [EphDem89]. Simulated Annealing has also been proposed for this problem [Paul85]. Here we discuss MLE.

We need the HMM parameter set H which maximizes the probability of generating the data in

$$X^{train} \times \{c_1, ..., c_m\}.$$

We assume that X^{train} is made up of sequences of observation vectors $y_r, r = 1, 2, ...$, and we search for H such that

$$H = \max_{H'} \prod_r P_{H'}(y_r|m_r),$$

where m_r is the correct model sequence corresponding to observation vector sequence y_r. In practice it is not possible to find the optimal parameters. A

gradient-descent procedure is used to iteratively converge to a local optimum in the parameter space; that is to derive a sequence of progressively better solutions $H_1, H_2, ...$

Take the case of discrete distributions. An HMM model is described by two parameter matrices, **A**, the state transition matrix of probabilities a_{ij} (probability of a transition from state i to state j), and **B**, the matrix of multivariate probability distributions. Since each matrix row must sum to 1, training is a problem in constrained optimization.

In each iteration of the Baum-Welch algorithm, parameter a_{ij} is re-estimated as

$$a'_{ij} = \frac{a_{ij}\frac{\partial P}{\partial a_{ij}}}{\sum_{k=1}^{N} a_{ik}\frac{\partial P}{\partial a_{ik}}}$$

[O'Shaughnessy87]. The same reestimation is applied to the entries of **B**. Theoretically the descent through parameter space is performed until $P_{H'}$ no longer improves between iterations; in practice only a few iterations are necessary.

6.3 Decoding Algorithm: Viterbi Search

Once the HMM parameters have been estimated, we need an algorithm which can use the models to decode the sequences. The decoding, or *recognition* problem is the search for the HMM unit-model sequence m_r which maximizes the conditional probability $P(m_r|y_r)$. The *Viterbi* algorithm provides an efficient approximation.

The Viterbi algorithm [Vite67] was introduced in 1967 for maximum likelihood decoding. It is a dynamic programming algorithm to find the lowest-cost path in a trellis, where the cost of a path at a given trellis node n_j can be computed as the sum of the cost at the previous node n_{j-1} and the cost incurred to get from n_{j-1} to n_j. Define the cost $C(t|m)$ of a path t through an HMM as -1 times the a posteriori log-likelihood of the path:

$$C(t|m) = -\log P(t|m)P(y_r|t).$$

Let $C_l(i)$ be the cost of the lowest cost path ending in state i at time l, and $C_l(t,i)$ be the cost of going from state l_t to state i at time l, using transition t. If t is full, then it comes from from l and $c_l(t,i) = -\log q_t$. If t is full, then it comes from time l-1 and $c_l(t,i) = -\log q_t b_t(\underline{y_l})$.

C_l can be computed as

$$C_l(i) = \max\left\{\max_{t.empty}(C_l(l_t) + c_l(t,i)), \max_{t.full}(C_{l-1}(l_t) + c_l(t,i))\right\} \quad (6.10)$$

The lowest cost path is the one with the highest probability. The Viterbi algorithm finds this cost by computing

$$\max_t C(t|m) = \max_t(-\log P(t|m)P(y_r|t)) = C_{l_y}(F), \quad (6.11)$$

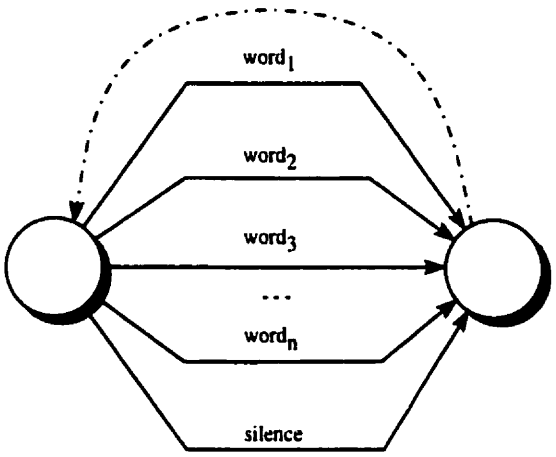

Figure 6.3: A looped word-recognition model.

where $C_0(0) = 0$. A trivial modification to the algorithm that computes $C_l(i)$ can provide the actual sequence of state transitions in the optimal path. All through the recursion, a *back-pointer* $B_l(i)$ to the transition that resulted in the best path "so far" is saved. At the end, the pointers are used to trace the optimal path from $C_{L_y}(F)$ to $C_0(0)$. The optimal state-path through HMMs is often used as an alternative to Equation 6.1 (for $P(y|m)$), because it is faster to compute.

Although the optimal state-sequence is the most likely path through the models, the sequence of unit-models corresponding to this path may not be the most likely models. This is because the probability of a model sequence must summed over *all* the paths in the sequence, and not only the most likely path. Usually, however, the best path provides a close, efficient approximation.

Consider creating a general model m_{gen} by looping the unit models together, so that the end state of the last unit model in each word has an empty transition to the start state of the first unit model in each word, including itself. Figure 6.3 shows a model for connected word recognition. In this case the most likely path through m_{gen} corresponds to a series of paths through individual word models. The Viterbi algorithm effectively provides a word sequence corresponding to an acoustic observation sequence y. Simultaneously, it *segments* the observation vector into subsequences corresponding to specifically recognized words. These words are the output of the recognition procedure.

6.4 Beam Search

In Viterbi decoding, quantity C_l of equation 6.10 must be computed at each time l for every node (ie. state) in the network which constitutes the language model m. The end result is the likelihood of the optimal path with respect to the exponentially large set of paths through the state-space. In the jargon of heuristic search algorithms, Viterbi produces an *admissible* (provably optimal) search.

The complexity is $O(|m| \times r)$ where r is the number of frames and $|m|$ is the number of states in the network. All though the algorithm is efficient in terms of finding the solution in time polynomially bounded by the size of the language model, in practice terms the algorithm fails because it spends too much time estimating the many low-likelihood paths through the network. Consider that in a large vocabulary ASR system, there may be thousands of words, each modeled by several phoneme HMMs, each of these containing several states. A practical language model may construct the network using n-gram statistics, inserting the same word in many parts of the graph depending on the conditional transition probabilities with respect to other words. As a result $|m|$ may be on the order of millions. For example, the AT&T system described in [LjoljeEtAl95] used a 60,000 word vocabulary with *34 million 1-5*grams. Since speech recognition must be performed online and in real time to be practical, the computational price of admissibility is too high.

The solution is to add a threshold computation to the Viterbi algorithm. For each frame l of the signal, the maximum state-likelihood is tracked as equation 6.10 is computed. Each state has a boolean flag indicating whether the state is *active*, initialized to false before the current frame is calculated. When the state calculations are done, each state s is set active if and only if

$$C_l(s) \geq \max_i(C_l(i)) - \theta \tag{6.12}$$

where θ is a parameter controlling the exhaustiveness of search. In frame $l+1$ only transitions from active states have their likelihoods computed. The difference in equation 6.12 describes a *beam* through the search space containing the most likely state-paths; this threshold-mediated Viterbi algorithm is called *beam search*.

The effect is to prune from the search space all paths whose cumulative likelihood is significantly less than the running optimum. This can be justified by observing that if the initial segment of a path is very unlikely with respect to the data so far, it is almost certainly the wrong path. Exhaustive search will compute the observation likelihood $P(\underline{y}|i)$ of state i even if the path probability $P(t|m)$ leading to i is nearly zero. For example, states close to the initial node of the network would be considered all through the procedure, even though they could almost certainly not have produced the features observed near the end of the signal. Beam search is more efficient and the

benefit is cumulative, since early pruning of low-likelihood states prevents the unnecessary overhead of extending these paths in subsequent frames. The complexity of beam search is $O(\overline{m} \times r)$ where $\overline{m}$ is the mean number of states entering the beam per frame. This mean depends on the beam threshold and is independent of the complexity of the language model.

Chapter 7

Classical Search Methods

This chapter discusses typical solutions to the search problem in automatic speech recognition, based on the underlying tools of the previous chapters. The speech programs of BBN Systems and Technologies may be considered one representative example. The BBN research system is called BYBLOS. It is a speaker-adaptive continuous speech recognizer based on context-dependent discrete-distribution hidden Markov models. The first published results, in 1987, were 93% word accuracy on a 1,000 word task of read speech.By November 1994 the system was reported to have achieved 96% accuracy on the 2500 word **ATIS** task, a corpus of natural-language queries to an airline travel information database. On the **Wall Street Journal** corpus, a collection of read speech containing 20,000 words, the system was reported to have achieved 89% word accuracy.

Unfortunately the performance of a recognition system on a carefully recorded test set composed of read speech does not predict how it will perform under realistic conditions. Spontaneous speech is filled with hesitations, non-verbal utterance, non-grammatical locution, and spurious environmental noise. Thus, performance badly deteriorated when the system described above was tested on the **Switchboard** corpus, a database of 143 hours of recorded spontaneous speech containing 22,000 words. (Weighted by frequency, 5,000 words cover 97% of the data.) Despite using an extensively trained statistical language model based on two million word-pairs, BYBLOS achieved a mere 50% word accuracy on the task.

BBN developed a commercial product based on BYBLOS which it called HARK. This real-time system was fully-software based - its only specialized hardware requirement was a 16kHz linear-sampled audio acquisition channel. The product has a basic module that processed continuous speech from a 2,000 word active vocabulary. By the spring of 1995 it was able to do dictation from a 40,000 word active lexicon.) The system switched between different active vocabulary/grammars specific to different tasks. Although speaker-independent to a degree, BBN emphasized the importance of adapting the acoustic models to its customers' voices in order to maintain a target accuracy of approximately 3-4% word error rate. After adaptation, it had the

ability to switch the active acoustic model set based on the target speaker using the system.

The systems described above are still representative of current speaker-independent ASR technology for continuous speech. They also illustrate the continuing weaknesses of current methods. Key features are:

- *Phoneme (triphone) HMMs.* These are supplemented with word models for short and common words.
- *A network representation.* The language recognized is represented as a finite state network of phonemes, in which legal paths correspond to words in the lexicon.
- *A pronunciation dictionary.* The phoneme sequence for each word is drawn from a pre-compiled dictionary.
- *A statistical grammar.* The network is a combined phrase grammar and statistical grammar based on word-pair probabilities (bigrams). The speech signal is fitted frame-synchronously, left-to-right, to the network to produce the optimal sequence.
- *Closed vocabulary.* The success of the system depends on being able to model speech utterances a priori. If a specific syntax is imposed, certain illegal word sequences may be generated and then rejected. However, there is no integrated way to spot and reject out-of-vocabulary words. The highest-scoring hypothesis wins.
- *Speaker adaptation.* The system performs on-line speaker adaptation. For best results the initial models requires adaptation to the user's voice.
- *Task dependence.* The system works better with a medium-size vocabulary and a grammar depending on the current assigned task. This reduces the scope of the search and enriches the context, improving recognition. Performance seriously degrades with larger vocabulary and less context.

7.1 The Classical Word-Lattice Approach

Another good example of ASR methodology is the SRI `Decipher` system [MurveitEtAl93]. This type of system used a multi-pass algorithm in which the first stage module is an acoustically-driven recognition engine. The acoustic module employs HMMs. Generally, a first-pass recognition module will have models for phonemes, often supplemented by whole-word models for short or common words. This module attempts to translate the signal into an accurate *lattice*, or time-segmented graph of word hypotheses.

The word lattice is generated one of two ways: the *network method* and the *lexical access method.* In both methods, the lexicon of valid words is generated from a dictionary in which the legal pronunciations of each word are represented by phoneme sequences.

In the network method, the lexicon is converted into a graph in which the arcs represent the transitions between phonemes. Each word in the lexicon corresponds to a path in the network. The language model consists of n-gram probabilities attached to the arcs between words. In this way, for example, the conditional probability of a common word-pair (or bigram) is attached to the arc connecting the two words in the network. If the language-corpus derived probability of word-pair is zero (as would be the case for many word pairs that are semantically possible but poorly represented in the training corpus) a transition between the two words is still possible through a special arc representing the *backoff probability estimate.*

The phoneme HMMs are linked together in the shape of the network. This large automaton then represents the language of the recognizer with all the acoustic, lexical, statistical, and grammatical elements integrated. A beam search is used to align the signal with the model and generate the lattice of hypothesized words with corresponding time denotations.

A consequence of using one automaton to recognize the whole language is the size of the resulting program. Even with a moderate vocabulary of $5,000$ words, the nodes in the network can number in the millions. The machine memory and CPU time required to process this kind of network is considerable, and any useful recognition system must be made to work in real-time. Consequently, researchers have looked for ways to compress the graph by eliminating as much redundancy as possible [AntoniolEtAl94].

Even compressed, the network model is too large for fast recognition if detailed acoustic models are used. For this reason the phonetic network is not used as a sentence hypothesizer in large ($> 10,000$ words) vocabulary ASR systems. It is used instead to generate a word lattice in a first-pass. The lattice is given as input to a second stage module which performs a more detailed analysis. The system described in [NeyAubert94] used this approach, generating a lattice with an average of 10 hypothesized words per spoken word, and then applying a trigram language model to this vastly reduced language space.

The second-stage recognizer may use more detailed acoustic models in fitting the acoustic signal to the word hypotheses in the lattice. For example, in [BocchieriEtAl95] the first pass used context-independent models with small mixtures. The word lattice was converted into a time-annotated reduced network, and bigram probabilities were embedded in the reduced graph. The second-pass employed a much larger set of allophone HMMS with large numbers of distributions per mixture. These more accurate models were used to re-estimate the maximum likelihood word string from the reduced network with respect to the already computed acoustic feature vectors.

Alternatively, the second-pass may avoid any more acoustic analysis, and

treat the word lattice as symbolic data on which to apply higher-level language constraints. These constraints can be the syntactic/semantic knowledge of a parser, or they can be language model statistics.

In contrast with the network method, the lexical access method uses the acoustic models to produce an accurate phoneme string, or alternatively, a lattice of phonemes. The word hypotheses are then generated symbolically, using the phonemes to access the lexicon and produce the most likely set of candidate words.

Since the phoneme space is much smaller than the search space in the network method, much more detailed and accurate acoustic models can be used in the initial stage of the lexical method.

ASR systems built with a multi-stage process incorporating LPC or Mel coefficients, a language network or phoneme network, bigrams word probabilities, and word lattices have been well established for years, and may be referred to as the *classical* ASR approach.

7.2 Sources of Error in Word Lattices

The classical method produces a word lattice in a first-pass search, and it is therefore crucial the lattice be as accurate as possible. If the correct word is not present in the word lattice, the second-pass, however accurate, cannot produce the right string. No system, however, could produce a word lattice with perfect accuracy, unless the search beam is widened to the point where nearly everything is hypothesized everywhere.

First, the acoustic analysis is imperfect. The feature vectors are produced by applying the Mel-scale transform to the signal spectrum. The Mel filters are logarithmically spaced triangular filters that have been derived to correspond to *critical bands* in the auditory model of speech perception. They are used because they usefully project the spectrum into a small-dimensional, perceptually significant feature space, and furthermore, they are used because they have been shown to be more effective than alternatives (when there is a perfect match between the training and test conditions.) They are imperfect because of the limitations of the auditory model on which they are based, and because the frame-based FFT computation that generate them must trade off the accuracy with which long and short-duration changes are captured.

A second, inevitable source of error is the pronunciation modeling problem. In continuously spoken English, the words are often distorted from their dictionary, or *canonical*, phoneme representations as a result of speaker tendencies (such as accents), contextual articulatory events, as well as random articulatory variability. ASR systems address this variability by using clustered context-dependent models, and incorporating multiple lexical "spellings" for each word in the dictionary.

There are two basic reasons for modeling phonemes rather than words at

the acoustic level. First, the number of acoustic models to train is fewer, and independent of the size of the vocabulary being modeled. This reduces the complexity of the acoustic module, and allows users the modify or expand the system's dictionary without retraining the acoustic models. Second, it is possible to collect adequate amounts of training samples from the available speech corpora because the basic units are all well represented in many contexts. But the problem of varying word pronunciations reasserts itself here at the lexical level: it is difficult to acquire *a priori* knowledge of how a given word may be pronounced by different speakers in different contexts. For any large vocabulary there will be many words whose pronunciation is poorly represented in the training corpora, and as a result only the canonical representation is present in the dictionary, perhaps with one or two possible distortions.

Another intrinsic limitation of the classical approach is particularly evident in the lexical access method. The acoustic analysis is performed in the first-pass, and the second-pass accesses a string or lattice of phonemes as symbolic data. As stated above, there will be always be instances in which the correct phonemes has been missed, and are simply not represented in the lattice. In this case there is no way for the recognizer to recover the correct hypothesis because the acoustic information has already been discarded. This problem exists for the network method, as well. Although the latter reprocesses the acoustic data, it uses are the same Mel (or PLP or LPC) based features as before and suffers the same limitations. Furthermore, the second pass involves a maximum-likelihood search of the reduced network produced by the first pass, and whatever was missed the first time cannot be re-introduced here. These problems are inherent to the systems described. What is desirable is some method of *rescoring* the signal on the basis of both knowledge gained in the first-pass, and additional acoustic tests, in order to improve the accuracy of the reduced network by *adding* hypotheses which were missed the first time.

7.3 Out-of-Vocabulary Events

Another limitation of the classical approach is the *out-of-vocabulary* (OOV) event. In a live ASR system, whether dealing with dictation or dialogue, the spoken input will reflect the real nature of human speech in containing numerous utterances which do not correspond to lexical items in the dictionary. These OOV events include non-dictionary words, hesitations and pauses, false starts and restarts, and *spontaneous speech utterances*. Spontaneous speech utterances are the non-verbal noises, the "ahs" and "ers" with which all real speech is liberally interjected.

The difficulties of modeling OOV events are plainly evident. The classical method relies on a best-fit strategy in which the likelihood of dictionary items occurring in some sequence is estimated *a priori* by the language model.

Spotting OOV events requires a strategy of *rejection*, or the system will inevitably produce an incorrect word sequence. But in order to spot the OOV event one has to model it accurately, or the chance increases of rejecting actual dictionary items with non-ideal pronunciations. It is difficult to model the OOV event accurately because of its inherent variability. Further, it is difficult to estimate the conditional probability of such events occurring in the context of other lexical items, ie. it is hard to incorporate the OOV event in the language model.

In the classical method, rejection of spontaneous speech utterances is attempted through the use of *garbage models*, *anti-models*, and *noise models* [WilponEtAl90]. The garbage model can be an HMM trained on spontaneous utterances, or on the average spectral events contained in non-silent portions of the training signals. The noise model would be trained on non-verbal utterances only. OOV words can be spotted using a looped network of phoneme HMMs. If the log-likelihood ratio comparing the winning hypothesis with the garbage hypothesis is beneath a certain heuristic threshold, the corresponding segment of the signal is denoted as an OOV segment, and rejected. Pauses and hesitations can be modeled with *silence models*, and these have to be incorporated in the language model network effectively, so that the pause can be "consumed" by the automaton wherever it might reasonably occur in the utterance.

False starts and restarts are common events in which some word or phrase is partially completed, the ending distorted or broken off into a slight pause or non-verbal utterance, and the utterance repeated whole from the beginning. This kind of event, simple to describe, is very difficult to model in the classical system. The lexically accurate portion of the signal is not rejected, but the complete utterance contains a repetition of word(s) which cannot be pre-estimated by any language model because of its inherent randomness. A grammar or parser would be able to reject a phrase that contains a restart. But this approach is ultimately undesirable for two reasons: the utterance that contains a restart is often intelligible and should not be rejected; also the eventual goal of ASR is to reliably transcribe speech that is ungrammatical, as this characterizes most utterances actually spoken by human beings.

Chapter 9 introduces a multi-pass search method based on syllable-like segments which is intended to compensate for some of the problems discussed in this and the previous section.

Part III

New Algorithms and Experimental Work

Chapter 8

Search for Acoustic Models

This chapter describes the development of a speaker-independent automatic speech recognition system using hidden Markov models (HMMs). Simulated Annealing and randomized search are used to optimize discrete features of the system, including topologies, parameter tying, allophone context clusters, and the sizes of mixture densities. Domain knowledge is used to initialize and to constrain the search, which optimizes recognition performance while reducing the number of model parameters. System performance results for new types of discrete and continuous HMMs measured on the **TIMIT** corpus are reported. The small set of context-independent phoneme HMMs produced is competitive with much larger systems of context-dependent models.

8.1 Introduction

An HMM is a probabilistic finite state automaton which can model a stochastic information source with a good compromise between simplicity and generality (Chapter 6). The list of states, connecting arcs, and the mapping which assigns to them a probability density function, are collectively referred to as the model "topology". While rigorous mathematical methods have been developed for estimation of acoustic parameters, the choices made for the topology, for tying distributions among different arcs or states, and for the phonetic or phonemic events to be represented by an HMM, have been arbitrary.

It is well established by now that having different HMMs for allophones of a given phoneme substantially improves the performance of Automatic Speech Recognition (ASR) systems [LeeEtAl90A], [KimballOstendorf92]. It is also well known that training a large number of allophone models requires a very large training corpus containing many samples of each allophone model to be trained.

Various training methods have been proposed to reduce the imprecision of parameter estimation if a limited amount of data is available for certain phones. One is to train models at different levels of detail—isolated

phonemes, and phonemes in the left and/or right context—and to interpolate the statistical parameters of these models in order to obtain the parameters of allophone HMMs [ChowEtAl86]. Various ways of tying the statistical distributions associated to different elements of HMMs have been studied [Young92]. Another method consists of clustering contexts, and performing a sort of generalization using classification and decision trees [Hon92], such that enough training samples are available for each cluster, and clustering repects some mathematically defined criteria about the "impurity" introduced when merging allophones. The decision tree method of deriving allophone states and their corresponding density functions has become perhaps the most popular tool for training acoustic models at the present time [BahlEtAl91], [Chou97], [BeulenNey98] [ChouReichl99].

This chapter proposes a methodology for a mathematically sound solution of a group of problems, some of whose solutions until now have usually been arbitrary. [1] The basic concept is that choosing a set of allophones, HMM topologies, and tying of distributions is a *search* problem. There are various methods and measures of success in directing a search toward some goal.

In Section 8.2, simulated annealing is proposed as a search method for the above mentioned problems, guided by the recognition performance on a subset of the experimental corpus disjoint from the test set. In general, search complexity is prohibitive, so suitable heuristics have to be used in order to constrain it. Section 8.3 describes how search is used to derive topologies and distribution ties. As well, heuristics based on speech knowledge are proposed to constrain the choice of phonemic and phonetic contexts which characterize a cluster.

The chapter reports experimental results for phoneme recognition on the **TIMIT** corpus using the allophone models obtained with the above-described search procedure. The recognition language model for our experiments is a loop of allophone models similar to those described in the literature [LeeHon89]. The experiments show small improvements obtained with topology optimization, and substantial improvements with allophone models.

It is well known that the effectiveness of HMM parameter estimation depends on the initial values assumed for the parameters. In principle it should be possible to use the improved topologies and allophones as starting conditions for the design of new and better phoneme models. Section 8.4 describes how allophone models corresponding to the same phoneme are merged into a single phoneme model. The performance of the new phoneme model is close to that of the allophone models, suggesting that the method proposed here allows one to build a small and effective set of phoneme HMMs that are competitive with a larger set of allophone HMMs.

Further improvements are obtained by conceiving simple HMMs, one for

[1] The experimental work in this chapter has been published by the author and R. Demori in similar form in volume 9, number 2 of the journal "Computer Speech & Language".

each phoneme, with mixtures having a large number of Gaussian distributions only at the beginning and at the end of the model. This choice is supported by the conjecture that coarticulation effects produce large parameter variability at the boundaries between a phoneme and its neighbours. The initial values of the parameters of the Gaussian distributions are the ones of the trained distributions in corresponding initial and final arcs of the allophone models. As well, improvements can be obtained by eliminating similar Gaussian distributions and retraining the models with a reduced number of mixtures.

Section 8.4 also shows how performances can be improved by introducing simple acoustic parameters describing time and broad-band features not well characterized by Mel-scaled cepstral coefficients and their derivatives.

8.2 Search Strategies

8.2.1 Simulated Annealing

In contrast with hill-climbing or gradient-descent optimization methods, randomized search does not set out to thoroughly explore the vicinity of a local extremum of the cost function, but employs instead a random solution generator to visit points all over the solution surface. One advantage of this approach is that it is easily applied to almost *any* optimization problem, continuous or discrete, regardless of non-linearities or discontinuities. A randomly generated set of solutions S_i is unlikely to contain the global or even a local extremum to a non-trivial cost function *f(S)*. However, there is a higher probability of yielding a solution S^* such that

$$|f(S^*) - f(S_{optimal})| < m$$

where m is some acceptable margin for error in the solution.

The goal here is to find improved values for certain discrete parameters of an HMM speech recognition system. These parameters include the *topology* of the unit models. The cost measure to be optimized is a measure of the recognition performance of the system. As an example, consider the problem space represented by an HMM topology restricted to topologies with no more than 7 states and 7 output distributions. The state–transition matrix has 49 entries, each of which can contain 9 values. (7 values for the seven possible distributions, plus an extra 2 values representing either *no* transition, or a *lambda* i.e. non-consuming transition.) This means there are 5.7×10^{46} distinct solutions. Clearly exhaustive search is infeasible.

Although the computational cost of generating and testing new solutions prohibits more than a cursory search of the problem space, a pure Monte Carlo search can nevertheless have practical utility. Unless the initial solution is a good local optimum, any series of randomly generated candidates

which are perturbations of the initial solution is likely to contain some candidates which improve the performance measure. Care must be taken with this approach, however, since the goal is to derive a speech model which generalizes well to new data. A more directed kind of search is feasible, in which a method is exists to escape from local minima. For the last fifteen years researchers have applied the technique of *simulated annealing* [KirkpatrickEtAl83] to many areas where gradient and other hill-climbing methods were unavailable or inadequate. Simulating annealing is a method of randomized optimization which acts like a hybrid between Monte Carlo search and hill-climbing, beginning like the former and settling into the latter near the end of the search.

In *Boltzmann annealing*, the solutions to a given cost function are assumed to be distributed with the Boltzmann probability factor $P(S_i) = \exp(-E(S_i)/kT)$ where $E(S_i)$ is the cost-function value for solution S_i, k is Boltzmann's constant, and T is a system parameter called *temperature*. A new solution is generated randomly, and accepted if the cost improves (ie. $\Delta E < 0$). Otherwise, the new solution is accepted according to the probability ratio

$$\begin{aligned} \frac{P(S_{i+1})}{P(S_i)} &= \exp(-(E(S_{i+1}) - E(S_i))/kT) \\ &= \exp(-\Delta E/kT) \end{aligned}$$

This conditional acceptance of non-improvements allows the search procedure to escape local minima. As T is reduced, the probabilities given by the Boltzmann distribution vanish for all but the lowest-cost solutions. It can be shown that, given the above assumptions, if the temperature is lowered in stages and enough solutions are sampled at each temperature, Boltzmann annealing will converge to a globally optimal solution. One "cooling" schedule that guarantees optimality is the logarithmic schedule

$$T_n = T_0 \frac{\ln n_0}{\ln n}$$

where n is the temperature iteration and $\{T_0, n_0\}$, are some reasonable starting values. In practice faster schedules are used which, while sacrificing the theoretic property of convergence, tend to provide useful optimization results. The latter approach is called *simulated quenching*.

In this research simulated annealing is used to optimize the structure of HMMs for English phonemes. Because of the prohibitive size of the search space, a variety of non-optimal schedules, including linear ones, were employed. During the annealing, new solutions were generated from old by randomly permuting some discrete value, such as the value of an entry in the matrix representing the HMM topology. The measured recognition accuracy of the HMMs provided the value for cost function E.

8.2.2 Knowledge-guided Search

The solutions searched for are refinements to the structure and organization of hidden Markov models for speech, things that are usually set by hand. The models arrived at are in turn fitted to training data in the form of acoustic speech samples. To ensure the things learned about HMMs are generalized improvements, the objective function that drives the annealing process must be evaluated accurately. This means re-training and re-evaluating the models on thousands of acoustic samples every time a new model structure is generated. The computational cost of this procedure effectively precludes exploring the search space very well. In topology experiments, for example, only a few hundred solutions were tested at each temperature. However, any measurable improvement on existing solutions is desirable.

Another interesting approach would be to let the amount of data used to train and evaluate the models serve as the system "temperature". This would allow more solutions to be tested during the early part of the annealing, with the higher error in the evaluation function serving as a probabilistic factor affecting the acceptance of new solutions. Then as the system "cooled," increased training would provide a more and more accurate measure of solutions, and direct the system more toward a better solution (in this case lower recognition error.)

In any case, since the problem space is large and the ability to explore it limited, the best way to solve the problem is to use the methodology in tandem with knowledge of the problem domain. Search begins with knowledge-guided and empirically proven solutions. Randomized search is used to develop new and better solutions. These are evaluated in the light of knowledge about the speech modeling problem, and adjustments are made. If necessary, the entire procedure can be repeated. In the following experiments, this approach of *knowledge-guided* random search has proved to be an effective compromise, and provided better models for speech.

8.3 Applying Search to HMM Structure

8.3.1 Basic Recognizer Architecture

In each experiment, the original models are first evaluated by training them on a set of sentences, and testing their recognition performance on another set. The train/test suite is later used to measure the optimized speech system, to see if recognition performance improved. The sentences used to score the models *during* search belong to a third, separate test suite. All acoustic data were drawn from the **TIMIT** acoustic-phonetic speech corpus. 3679 sentences were used for training, and the core test of 192 sentences containing 7,333 labelled phonemes was used for final evaluation.

Both discrete and continuous HMMs were trained and evaluated. For the continuous models, the HMM output distributions of 12 Mel-scaled cepstral

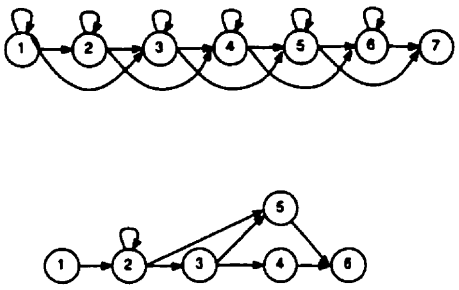

Figure 8.1: The initial (top) and optimized topology for model "b".

coefficients, 12 difference cepstrum coefficients and energy and energy rate-of-change, were estimated from the training data using continuous mixture densities composed of Gaussians. The covariance matrix is assumed to be diagonal. For the discrete models, three codebooks, one for each feature set, were introduced (256 entries were used for each of the first two sets and 64 for the energy features.)

To speed up search, continuous models were abandoned in favor of discrete ones during optimization experiments. The HMMs were trained by Baum-Welch re-estimation. Recognition was performed using Viterbi maximum-likelihood decoding over a looped, phonemic finite state network.

8.3.2 Topologies and Distribution Ties

As mentioned in the introduction, the topology and tying of distributions in hidden Markov models for speech are usually determined *a priori*, or based on statistical measures [Young92]. Some researchers have made attempts to evolve the topologies algorithmically [Casacuberta90EtAl], [SanchisCasabuberta91], [JouvetEtAl91], [TakamiSagayama92]. In the method described here, the topology and distribution assignments of an HMM are represented as an integer array, and simulated annealing is used to optimize this data structure. Each perturbed solution is applied to the basic English phoneme models which are trained on one target set of sentences from the **TIMIT** database, using discrete parameters. The models are then tested on a separate test suite, with the measured unit accuracy used as the objective cost function to be maximized.

After a set of topologies adapted to particular phoneme classes have been derived, they are evaluated on a third data set of 192 sentences. Figure 8.1 shows one such topology. A mix of the best performing topologies are selected (based on individual phoneme accuracy,) and finally evaluated by training a model set with continuous mixture-densities.

The searches are initialized with a proven, left-to-right model, and limit the growth of the model to 7 states. Using domain knowledge in this way, the

continuous models	no. distr.	plosive errors	vowel errors	phone error rate
7 state topology	1440	402	924	44.06
mixed topologies	1362	385	890	43.62

Table 8.1: Comparison of initial and optimized topologies.

search is constrained enough to achieve a reasonable solution in reasonable time. Since the different phoneme classes are acoustically dissimilar, it is conjectured that different topologies should be developed for specific broad phoneme-classes.

Search began with the initial topology represented by the top of figure 8.1. The optimization proceeded by repeatedly perturbing values in the transition matrix, and evaluating the new topology represented by these perturbed values. Since the values represented both transitions and the distributions tied to them, the method optimizes simultaneously over both topology structure *and* parameter- tying.

48 or 53 units are modeled, depending on the experiment suite. Following [LeeHon89], the original unit-models are mapped onto a simpler set of 39 recognition units for classification. All performance results in this chapter are for these 39 units. Recognition performance is given by the *phone error rate* (PER):

$$PER = 100 \times \left(\frac{\#insertions + \#deletions + \#substitutions}{\#units} \right)$$

Topology-optimization was performance-based, using 512 sentences to train and 96 sentences to evaluate each perturbation of the topology set. The success or failure of the experiments was determined by testing models on another set of 96 sentences. After many stages of optimization driven by large-sample performance, a best set of class-specific topologies was derived. This set contained seven new topologies for the phonetic classes *silence, closure, fricative, nasal, liquid, vowel,* and *plosive.* The models were trained as many iterations as possible until performance failed to improve, and then tested on a new data set. A unit-by-unit study was made of how the new topology performed relative to the 7-state topology, counting the number of errors E associated in each case with a given phoneme p. E has four components: *insertions, deletions,* the number of times a p is misrecognized as something else, and the number of times something else is misrecognized as a p. Using the measure E, a new model set was constructed, a mixture containing the better-performing topology for each phoneme. This mixture showed an improved performance with a reduced number of Gaussian distributions (Table 8.1). The mixture showed a 2% improvement in the discrete case, and 1% in the continuous. The mixture was even better when considered on a per-class basis, significantly reducing the number of errors for vowels and hard-to-distinguish plosives.

8.3.3 Allophone Context Clusters

The same performance-driven optimization approach was applied to the problem of efficiently clustering contexts for allophone models. Since the problem space is large and expensive to search, acoustic-phonetic reasoning was used to determine a sensible initial grouping of contexts, and then simulated annealing perturbed these clusters so as to improve the recognition accuracy of models trained on these contexts. The output of the search procedure was also manually adjusted to conform to speech knowledge.

This approach is somewhat different from the tree-clustering algorithm of [BahlEtAl91], or the state-splitting approach of [TakamiSagayama92]. In this case, search is driven by performance and knowledge. In previous sections HMMs model the context-independent phonemes. Because of allophonic variability, better results are possible when phonemes are modeled in context [SchwartzEtAl85], [LeeHon89]. However, there are 48 possible *left*-contexts and 48 possible *right*-contexts for each phoneme. If each context of each phone were modeled separately, it would be necessary to train $48^3 = 110,592$ different models. In fact, the task would be difficult even with large speech corpora, since most of these contexts occur too rarely to be well represented.

The solution is to combine phonemic contexts into *clusters* which have similar contextual effects on the preceding or following phonemes. To model left-context units using any of 10 clustered contexts, the allophones may be represented with fewer than 500 models, a manageable number for which there are likely to be adequate training samples.

The problem is to choose the clusters appropriately. One approach would be to choose as fine-grained a context as can be well-trained from the available data. The more varied samples there are of a given speech unit, the more specialized models can be made for that unit. [KimballOstendorf92] suggest a distribution can be adequately estimated when the number of samples is about six times the dimension of the observation vector. If the vectors contain 24 cepstral derived coefficients (Mel and ΔMel), at least 144 samples per model are needed. Depending on one's point of view, a rigid threshold like this may result in too many or too few models. (In practice, good results are achieved with some units trained on fewer samples.) In [LeeEtAl91] another data-driven approach is used to select clusters, Following a *unit reduction rule* based on the number of samples available for a unit in the training data, they build a set of models containing 47 context-independent phones, 134 diphones, and 1101 triphones. While this approach guarantees the trainability of the units, it may produce a large number of models with similar distributions, in effect adding parameters without reducing system entropy.

In [LeeEtAl90A] an initial set of context-dependent models is trained, and these models are merged into clusters, called *generalized allophones*, according to some algorithm. The first method, *agglomerative clustering*, is

based on an entropy distance measure applied to the allophones. The second method is a heuristic decision tree, in which the root consists of the complete set of allophones corresponding to a particular phoneme, and the leaves contain generalized allophones. At each node, the allophones are recursively divided into two sub-clusters based on the answer to a question provided by an expert linguist, a question designed to capture contextual effects. The recursion ends according to a metric of the distance between a parent cluster and its subclusters. In both above methods, the metric used to merge or split clusters is the "entropy increase" or information loss in the output distributions when two models are merged. Clustering is chosen so as to minimize entropy gain [Hon92]. A context-decision tree is also constructed in [BahlEtAl91], but their subtrees are divided according to a Poisson-model likelihood of the possible splits measured against the training data. The generation of subtrees is statistically dependent on not just the adjacent (left or right) contexts, but on the several preceding and following phones as well. In [RabinerEtAl89] two methods are suggested which proceed from the opposite direction: beginning with a generalized word model, they iteratively generate more HMMs to model that word, in which each HMM is re-estimated from different subsets of the training samples for that word. Although they don't address context, the same methods could in effect derive context clusters automatically, without appealing to acoustic-phonetic reasoning. However, the resultant context-sets would probably overlap. The state-splitting algorithm of [TakamiSagayama92] optimizes automatically along both topological and contextual axes, based on measured likelihood on the training data. A trivial Markov model is iteratively grown into a more complex model in which contexts are clustered and integrated. Jouvet [JouvetEtAl91] avoids the clustering problem, instead reducing the parameter space by integrating all left and right contexts into the topological structure of the allophone model. This can be seen equivalently as tying the distributions of the internal states of the allophone models.

In the experiments of this chapter, acoustic-phonetic reasoning is combined with the performance-driven randomized search described earlier in order to optimize the context clustering with respect to the available training data. In contrast with [TakamiSagayama92], recognition accuracy rather than training-set likelihood serves as the objective function. We call this approach *performance and knowledge-guided search.*

Optimization was first applied to the problem of efficiently clustering right-contexts for plosive allophones. Speech science suggests that the right-hand event, or succeeding phone, mostly affects plosive burst and transitions of relevant acoustic parameters. Intuition suggested an initial grouping of vowel right-contexts as shown in Table 8.2, in which some phonemes starting with the same symbol were grouped together. Simulated annealing perturbed these clusters so as to improve the recognition accuracy of models trained in these contexts. The output of the search procedure showed a tendency toward grouping phonemes by place of articulation. Minor corrections to

1. ao aa ay aw ax ah
2. ix
3. ae
4. ih
5. uw uh
6. er
7. iy
8. oy ow
9. eh
10. y
11. ey

Table 8.2: Initial right-context clusters for plosives.

1. ao aa ay aw ax ah
2. ix ih iy y ey
3. er ae
4. uw uh oy ow
5. eh
6. f v
7. s z
8. l
9. r
10. w

Table 8.3: Optimized right-context clusters.

clustering were manually performed to finalize the trend, resulting in the grouping shown in Table 8.3; some consonant clusters were also manually added.

Performance was used to drive the annealing. Since plosives were the model of interest, the number of errors on the plosive data alone served as the performance measure (ie. how many plosive events were deleted or misrecognized.) Almost all plosives in the training data are found in the the contexts of Table 8.3.

Optimization was next applied to the problem of efficiently clustering left-contexts for vowels. The initial grouping (Table 8.4) puts all the unsonorant consonants having similar place of articulation in the same class. The clusters produced after search are presented in table 8.5. The number of contexts was reduced from 23 to 13.

A summary of results for discrete models using various types of context-dependent allophones is shown in Table 8.6. The solutions from all the searches were combined, and tested on new set of 96 sentences using discrete-codebook HMMs. The left contexts were expanded to include other consonant classes, using the context-clusters of Table 8.5. Next, the left-context

1. p b f v m	13. ax
2. s z zh th dh jh t d n ch	14. ix
3. ng hh g k	15. ih
4. l	16. ae
5. w	17. ah
6. ao	18. uh
7. aa	19. oy
8. uw	20. iy
9. er	21. ow
10. ay	22. eh
11. ey	23. sil epi bcl dcl gcl pcl tcl kcl
12. aw	

Table 8.4: Initial left-context clusters for vowels.

1. p b f v m
2. s z zh th dh jh t d n ch
3. ng hh g k sil epi bcl dcl gcl pcl tcl kcl qcl
4. l uh aw ow uw
5. w
6. eh er
7. ao
8. aa
9. ay ey ih oy
10. ax ix
11. ae
12. ah
13. iy

Table 8.5: Optimized left-context clusters.

discrete models	plosive errors	vowel errors	misrec. rate	phone error rate
48 phonemes	199	508	42.88	47.06
359 models left contexts	202	461	40.25	44.51
364 models left & right contexts 7-state topology	192	465	39.63	44.35

Table 8.6: Development of the recognizer with discrete HMMs. Phone error rate includes insertion errors.

continuous models	num. parameters	misrec. rate	phone error rate
48 phonemes	74,880	38.36	44.06
364 allophones left & right contexts	473,304	34.32	40.76

Table 8.7: Results for continuous models (no language model).

vowel and consonant models were combined with the right-context plosives. As hoped for, higher accuracy by class averaged out to a higher overall unit accuracy. Finally, these same context sets were used to train models with topologies based on the earlier optimization experiments. The results are in the last row of Table 8.6.

Results for the continuous-parameter models are summarized in table 8.7.

8.4 Derivation of Phoneme Models from Allophone Models

8.4.1 Merging and Retraining

To this point, the result of the search process was a set of unit model topologies and output distributions well trained for units in left or right context. The final experiment attempted to simplify the speech recognition system by merging the distributions of the allophones into phoneme units with parallel transitions. This was done for both discrete and continuous-distribution models.

In the discrete case, the n allophones for unit U were merged in a straightforward way. All allophones for U had the same topology $\mathbf{T}$. A new topology was created with n parallel transitions for each single transition in $\mathbf{T}$. Each of these parallel transitions was tied to the corresponding distribution in one of the allophones for U. Once these *merged*-model phonemes were built, they

discrete models	plosive errors	vowel errors	misrec. rate	phone error rate
53 phonemes merged dists	173	489	38.90	43.37
53 phonemes simplified	157	464	38.56	43.43

Table 8.8: Merging the discrete HMMs.

continuous models	num. parameters	misrec. rate	phone error rate
53 phonemes simplified	356,304	34.8	40.0
53 phonemes, with additional features	397,416	33.8	39.7
53 phonemes, fig. 8.2 topology	301,252	32.7	38.2
same, with bigrams		30.8	35.5

Table 8.9: Merged continuous context-independent models.

were then re-trained for four iterations. These models had a better phone error rate than the best allophone models (Table 8.8).

In order to simplify the models and further reduce the number of parameters, all the parallel transitions but one of the internal states of the merged models were removed. We hypothesized that extra distributions are only useful for modeling the contextual effects at the left or right end of the units. In fact, the simplified models showed improvements, after four training iterations, with respect to their predecessors.

The end result of the search process was to construct a set of discrete models with internal topological structure complex enough to model contextual effects of neighboring units significant to the particular unit. Results are summarized in Table 8.8. It appears that a substantial performance improvement can be obtained in context-independent models by just adding context-dependent distributions on the initial and final transitions of phoneme HMMs.

The same merge/simplify procedure was applied to the continuous models. Continuous-distribution HMMs already employ density mixtures. In this case the process of combining allophone distributions essentially means selecting mixture sizes for context-independent models and initializing the corresponding distributions with well-trained values. Results are summarized in Table 8.9.

8.4.2 Further Improvements

A constant avenue of research is the hunt for new acoustic parameters which are likely to contain information not well represented by MFCC, or other established feature sets. In the experiments described in this chapter simple new measures in the time domain and in broad frequency bands were investigated.

Following [DemoriEtAl76] the signal energy can be described in terms of peaks and valleys. Let an "event" be a peak or a valley. Let t_{b_j} be the beginning time of the j-th event $e(j)$. A temporal feature

$$temp(t) = t - t_{b_j}$$

is computed where t_{b_j} is the beginning time of $e(j)$ that covers a time interval including t. A value $\mu(t)$ is then computed as follows:

$$\mu(t) = \begin{cases} temp(t) & \text{if } temp(t) < \mu_0 \\ \mu_0 + \log_2(temp(t) - \mu_0) & \text{otherwise} \end{cases}$$

where μ_0 is a constant chosen a-priori. The feature $\mu(t)$ suggests the position of the t-th frame in the suprasegmental acoustic event t belongs to.

Two other features are obtained by introducing $e_1(t)$, the energy of the highest spectral value in the 100–900Hz band at time t, and

$$\Omega_{31}(t) = \log_{10} \frac{e_3(t)}{e_1(t)}$$

where $e_3(t)$ is the highest spectral value at time t in the 3–5kHz band. Performances were further improved by adding the new acoustic features, as shown in Table 8.9. Simple broad-band acoustic parameters and temporal features have the predicted significant positive impact on recognition.

A final experiment was performed using the same model topology, shown in figure 8.2, for all complex phoneme models (vowels, liquids, nasals, and plosives.) In this topology, the transitions represented by thick lines are modeled with a moderately large number of Gaussian distributions, while the transitions represented by thin lines are modeled with small mixtures. The relative sizes of these mixtures reflect the number of contexts in which the allophones described earlier were trained.

The initial distributions were taken from the well-trained merged-models described in section 8.4.1. These distributions were duplicated or reduced in number, so that each transition from the first state of the new topology could be tied to a mixture of 39 densities, each mixture tied to the fourth state could have 30 densities, and all the internal "thin" distributions could be mixtures of 6 probability density functions. The new models were then retrained for five iterations, and low-probability transitions were pruned. The results (third row in table 8.9) confirm the importance of good initialization of the parameters before estimation.

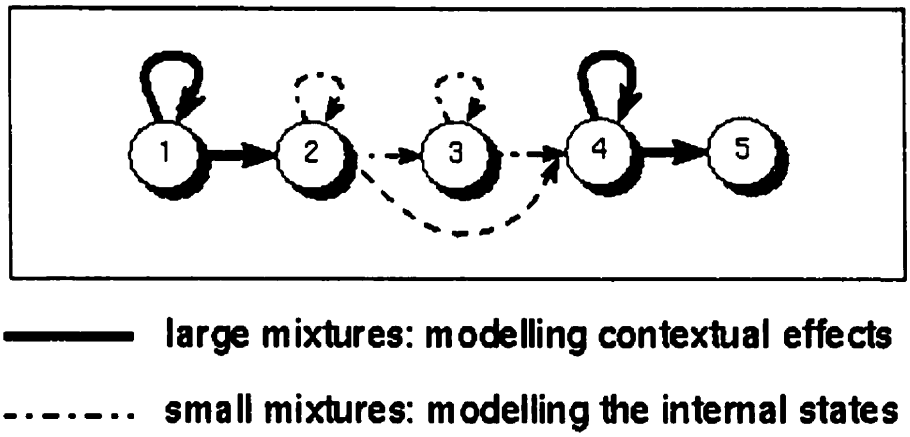

Figure 8.2: Final topology.

8.5 Discussion

Taking into account the comparatively modest feature set employed here, and the context-independence of the final models, these results compare favorably with those reported in the literature. The only systems reporting better phone-recognition rates on **TIMIT** around the time this research was published ([Robinson91], [YoungWoodland94]) employed 2nd-order derivatives and various kinds of context-dependency learning, e.g. allophone HMMs, or recurrent neural networks. The results described here indicate that a simple phoneme model containing rich mixtures of Gaussian distributions is a serious candidate architecture for certain applications; for example, the fast-match first pass of a large vocabulary ASR system. (Having a large number of Gaussians per mixture does not imply a high computation time if the number of mixtures is limited.)

In this research allophone models were initialized with the distributions of context-independent models, and vice-versa. From a mathematical standpoint it is better to perform an interpolation of new specialized models with the more general, better trained phoneme model from which they are derived. This should improve performance.

Chapter 9

Adding Knowledge with Syllable Phonotactics

A search technique incorporating the automatic modeling of lexical variability is introduced for speaker-independent speech recognition. Current state-of-art systems depend on being able to model the entire language based on acoustic features and the constraints of syntax or inter-word probabilities. These methods often fail in the presence of multiple speakers, new vocabulary, noise, and spontaneous speech phenomena. A new approach for word hypothesization is proposed, based on an acoustic-phonetic unit called the pseudo-syllable segment. An algorithm is described for transforming a sequence of syllables into words. Techniques are suggested for controlling the accuracy of the syllabic hypothesis set, and learning the phonotactics of syllables automatically in a statistical

9.1 Introduction

The output of an Automatic Speech Recognition (ASR) system fed by a speech signal is a sequence representing the words uttered by the speaker. The output of a Speech Understanding System (SUS) is an abstract conceptual representation of the meaning of the words uttered by the speaker. In both cases, words are hypothesized by a search process that produces the most plausible output according to a given scoring method for competing hypotheses.

The most plausible output depends on the Knowledge Sources (KS) used for driving search and may not correspond to the information conveyed by the signal. A search algorithm that always finds the best hypothesis is said to be admissible. Errors in the best hypotheses are due to imprecision of the KSs.

In theory and in practice, it is important to minimize the difference between the output of the ASR or the SUS and the speaker intention. With this perspective, search algorithms which are non-admissible because they

use sophisticated KSs may have better performance, in terms of recognition or interpretation, than admissible ones using simpler and less reliable KSs.

Admissible algorithms may be impractical with certain KSs due to time and memory constraints. An SUS or an ASR has to run in close to real-time and, more important, has to use an amount of memory within the limits of real technology. Actual non-admissible search algorithms use pruning thresholds during search. Candidate partial hypotheses with a score below a threshold are discarded before completion. Some of those hypotheses may raise their relative score and even become the best ones if allowed to reach completion. Reducing the selectivity of this process results in a increased amount of computation per spoken sentence and in an increase of central memory required. Technology is making available faster processors and larger memories, but since large vocabulary systems continue to increase the number and sizes of knowledge sources coupled to the search process, technology alone will not obviate the need to achieve real-time performance by trading off admissibility for speed. Furthermore, there are algorithms based on long-history language models (LMs), such as island-driven parsers, which have to work on already generated word hypotheses in order to avoid intractable theoretical complexity [CorrazzaEtAl91]. Generation of word hypotheses is an intermediate search phase in which hard decisions are made, ruling out candidates that a successive parsing stage could rescore as most promising. This approach adds motivation for the use of non-admissible search algorithms.

As the objective of ASR or SUS search is not admissibility, but performance, it is worth investigating non-admissible search algorithms, based on a cascade of processes with each process applying one or more specific KSs in a multi-step search. Most of the popular approaches so far use a language model (LM) to generate lattices or sub-vocabularies. LMs are obtained by analysis of large corpora of text and result are biased by the topics covered in the corpora and the preferences of the writers. In general the LM of a speaker is not the one derived by the analysis of available corpora. Even the lexicon is often not completely known.

For these reasons it appears useful to consider a way of generating a sub-vocabulary of words that can be present in a spoken sentence, starting from units that are not topic nor even vocabulary dependent, but are a complete set of basic units of a given language. Good unit candidates are phonemes and syllables.

It is possible to characterize a complete set of syllables in a language and drive the generation of syllabic hypotheses with a KS made of a network of all the possible syllables in the language. This network is complete because it represents all possible sequences of what can be said in a language and has a size that is tractable in terms of computation time and memory requirements. Furthermore, it introduces more constraints on the search space than a pure network of phonemes.

With good syllable models, it is possible to generate a lattice of syllable

hypotheses about a spoken sentence. This lattice can be pruned with additional segmental acoustic analysis and be used to generate word hypotheses. It is also possible to detect from the lattice hesitations, corrections, and other spontaneous speech phenomena, and to find which alternate pronunciation of a word is likely to have been used by the speaker. Word hypotheses may belong to a sub-vocabulary and be used to predict other sub-vocabularies using analysis of word associations. Words from the new sub-vocabularies can then be hypothesized from the syllable lattice.

In practice, the value of syllable lattices for sub-vocabulary generation can evaluated based on:

1. Word density, the ratio of the sub-vocabulary size to the number of words of the spoken sentence,

2. Word error rate (WER), the probability that a pronounced word is not in the sub-vocabulary,

3. Recognition accuracy in terms of word error rate of the complete search process.

9.2 Using Syllable Lattice or Sequences for Word Hypothesization

Another application of the syllable approach is to dictionaries. Current techniques for managing speaker variability include averaging the acoustic parameters over large training corpora, and introducing multiple lexical representations into the system [LjoljeEtAl95], [DemoriSnowGaller95]. Inspection of the causes of errors in classical, left-to-right word-fitting algorithms reveal that often the correct word is not introduced into the maximum likelihood sequence, or into the word lattice, because certain phonemes used to represent the word in the dictionary have not been articulated. Any attempt to re-estimate the acoustic parameters for the canonical phonemes on this data leads to a corruption of the model.

Adaptation at the lexical level is the correct way to compensate for this variability. New pronunciations may be learned through observation, or developed by linguists, but this may be impractically slow work for the development of new ASR applications. Some method is needed to generalize about the kinds of distortions typically produced with respect to canonical pronunciations, and generate new pronunciations automatically. This chapter proposes such a method. [1] In contrast with [DemoriSnowGaller95], this method hypothesizes different pronunciations at run-time, as an knowledge source integrated with the search procedure.

[1] The algorithms described here have been previously published by R. Demori and the author in the 1996 ICASSP proceedings.

The proposed algorithm uses a Viterbi decoder to generate a lattice of syllable-like units as a first stage preceding the generation of words. Alternatively, the n best sequences of syllabic segments can be generated. These segments consist of consonant clusters terminating in a vowel, and are generated in the standard way using a syllabic grammar with syllable bigrams, and backoff probabilities for uncommon sequences.

The pseudo-syllable is a good intermediate unit from which to carry out word hypothesization. Although phonemes may be deleted as an effect of speaker habit in continuous speech, some form of a syllable, however shortened or distorted, is likely to be detected if the underlying word has been articulated in an intelligible way. A study of the **Switchboard** corpus, for which the phonemes were transcribed by hand, revealed a 12% deletion rate for phones compared with less than 1% for syllables [GanapathEtAl97]. Also, with syllable data or smaller subword units, one can reason about the underlying utterance in a way not easily performed when the only information is a list or lattice of possible words. In the latter case, there is little chance for recovery if the correct word has been deleted from the list of candidates. Finally, the longer acoustic context of the syllable makes it more suitable than smaller units for exploiting parameter trajectories and other longer duration temporal and spectral variations.

Word hypothesization from syllables has been explored by some researchers. In [DentonTaylor92] the approach is discarded because syllabification errors cause a performance drop when the system is scaled up to larger vocabularies. In contrast, the system described in [LjoljeEtAl95] uses a syllable-graph generator in a multi-stage system designed to constrain the search space for very large (60,000 word) tasks. Both methods lacked explicit ways for managing syllabification in the presence of variable pronunciation.

The method introduced here is based on fitting the set of approximately 10,000 English syllables to the signal in order to produce a lattice of hypotheses segmented by time. It is necessary to first define the syllables, and then produce a reliable method of translating words of the task vocabulary into sequences of these syllable.

For simplicity we define a unit called the *pseudo-syllable segment* (PSS) [DemoriEtAl95]. It consists of zero or more consonants followed by a vowel. The PSS must be a legal sequence in the modeled vocabulary. For completeness of the word-PSS translation algorithm, *word fragments* are introduced to model consonant cluster word-endings.

The choice of these units is motivated by some particular advantages it offers to the segmentation behavior of the result lattice. A segment ending in a vowel will be characterized by an internal rise and steep descent in total signal energy. The corresponding portion of lattice can be re-scored by an acoustic measure specifically defined to disambiguate vowels. A lattice segment can be pruned of probable errors involving plosives or fricatives. The plosive is easily characterized by a near-zero drop in signal energy, corresponding to the glottal stop, followed by a sharp explosion of energy characterizing

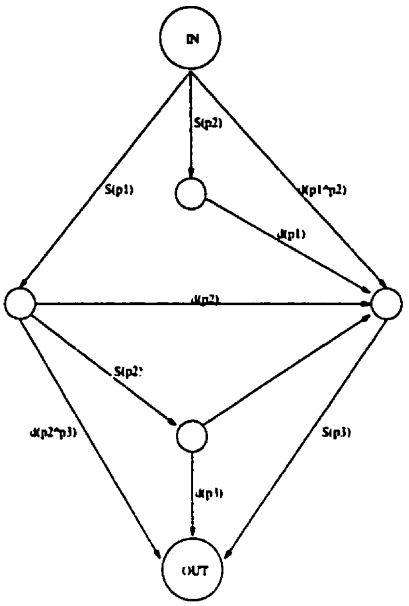

Figure 9.1: A syllable phonotactic model

the release. The fricative is characterized by a distinctly high ratio of high-frequency to low-frequency spectral energy. These measures can be used to post-process the PSS lattice by removing probable errors introduced by the Mel or PLP features. They can also *re-introduce phonemes that the HMMs missed.* This algorithm to improve on the weaknesses of the first-pass is a departure from and potential improvement on the classical approach.

A lattice segment containing a word fragment can be used to spot word endings. The lattice-to-word mapping of the second pass should be constrained so that the phonemes of these fragments are matched to word endings alone. Rescoring can be applied to fragments as well, in order to increase lattice accuracy, and eventually the completeness of the word network generated in the second pass.

9.3 The Phonotactic Model for Syllable-Mediated Search

Word generation is performed with a kind of model called the *syllable phonotactic model.* Figure 9.1 gives an example of a three phoneme syllable model. The phonotactic model is a hidden Markov model which contains statistics about phoneme symbols only, ie. lexical, not acoustic, information. The statistics $\overline{S}(p), d(p)$ model confusions of phoneme p within the context of a given syllable. $\overline{S}(p)$ is a discrete probability distribution modeling the substitution probabilities for phoneme p. $d(p)$ is the scalar probability of the

symbol being deleted during enunciation of the syllable. This model assumes that the beginning or ending phone of the syllable may missed but not both. The model can be enriched by adding an extra parallel state-transition path for each of the constituent phonemes tied to the probability distribution of phoneme insertions in that stage of the syllable. $d(p_i)$ constitutes the transition probability for the associated transition. Path transition probabilities are normalized by associating to transitions with a symbol distribution function $\overline{S}(p_i)$ the transition probability $1 - d(p_i)$

9.3.1 Initialization

Probabilities of substitutions, insertions and deletions are initially estimated in a context-independent way based on frequency counts from a labeled corpus such as **TIMIT**. A beam search is performed on the data set in order to generate a phoneme lattice for each signal and compare it with the correct labels. For each symbol i a frequency count f_{ij} is computed each time the symbol j is found in the lattice in a segment of the signal corresponding to i. Use of the phoneme lattice rather than maximum likelihood sequences provides a richer set of substitution counts reflecting both weaknesses of the acoustic models and pronunciation variability.

The phonotactic model for each syllable in the lexicon is initialized using these global statistics as follows:

$$s_i(p_j) = \begin{cases} 1 - \lambda & \text{if } i = j \\ f_{ij} \times \lambda & \text{if } i \neq j, f_{ij} > 0 \\ \epsilon & \text{if } f_{ij} = 0 \end{cases} \tag{9.1}$$

$$d(p_i) = 1 - \lambda$$

where f_{ij} is the normalized substitution frequency count obtained above for phone p_j with respect to ideal phone p_i. λ is a heuristic factor designed to give initially greater probability $s_i(p_i)$ to the canonical phoneme than to deletions or context-independent substitutions. ϵ is a small quantum to allow substitutions not observed in the training data.

9.3.2 Training

The syllable models may be trained using the following simple technique. As training data are made available, syllable-dependent statistics are averaged with context-independent ones weighted by the relative proportions of the data. First, the word transcription of a training sentence is converted to an idealized (canonical) syllable transcription. Then, the phonotactic models corresponding to this sequence are fitted to the syllable output of the phoneme-based decoder by computing the optimal Viterbi alignment of the

models to the observed syllables. By using the ideal syllable transcription, rather than the syllables actually pronounced according to **TIMIT** or some other hand-labeled corpus, the phonotactic model is constrained to our purpose: to learn about lexical distortion of syllables in speech, rather than to just model acoustic-phonetic variability at a different level.

If the syllable observations consist of a lattice, this lattice is converted to a network, and the models are aligned with the network. If no fit is possible (for example, the length of all n best sequences is different than the target sequence) the sample is discarded. For each observed syllable, and each phoneme in the correct phonotactic model, a counter is incremented for the inclusion, substitution, or deletion of the canonical phoneme according to the observation. After all data have been observed, these new frequency counts f'_{ij} are normalized and used to modify the values of equation 9.1 as follows:

$$s'_i(p_j) = \ (1 - \alpha(c(n)))s_i(p_j) + \alpha(c(n))f'_{ij}$$

where the $c(n)$ is a count of how many instances syllable n appeared in the training set, and the update weight α depends on the number of training instances. Then the procedure is repeated until no further changes in the alignments are seen.

9.4 From Syllables to Words

9.4.1 Synchronous Phonotactic Search

The phonotactic models are applied to the syllable sequences in order to produce scored word hypotheses. The method used is the standard Viterbi beam search. The output likelihood of this procedure is a measure $P(W_{jk}|W_j)$ of the degree of distortion exhibited with respect to the canonical representation of the scored word W_j.

The global search procedure associates to word hypotheses the score $P(W_{jk}|W_j)Pr(Wj)$, that is a combination of the language model and word distortion probabilities. It finds at the same time the most plausible pronunciation.

Once the word lattice is generated as described, interpretation can take place from the lattice using a scoring system that scores concepts and relative acoustic evidence for associated words. Dictation is performed by finding the maximum likelihood sequence among words that survive the first stages of search. The best sequence is based on a combined measure incorporating n-gram statistics, the acoustic scores for the words in the lattice with the hypothesized pronunciations, and the lexical score produced by the phonotactic models. This is done in a pipeline of three processes: syllables produced using simple acoustic models, words hypothesized with phonotactic models, and best sequence realized using detailed (triphone) acoustic models.

An advantage of the new approach is that unlike fast-match algorithms it uses an essential computation (the lexical distortion measure) to prune the initial search space.

9.4.2 Some Experiments

Experiments were conducted to measure the usefulness of word lattices produced from syllable hypotheses in the framework of the previous section. A standard word lattice was generated by a single pass decoder, using context-independent phoneme HMMs (described in [DemoriGaller95]), word bigrams and beam search. Different size beams were tested in order to provide an ROC curve describing the tradeoff between lattice density (mean number lattice entries per actual word) and global error rate. A similar curve was then produced using the syllable-phonotactic approach. This two-pass method employed the same phoneme acoustic models to generate a sequence of syllables under the control of a syllable bigram network, and then applied the symbol-generating phonotactic models to the syllables in order to generate a lattice of words. The latter pass was mediated with the same word bigrams as in the standard method. Results indicated that the density and WER were higher for syllables than for the standard lattice. For better results more work is needed to improve the training of the syllable phonotactic models.

9.5 Filtering the Syllables

A major weakness of the standard algorithms is the inability to hypothesize the correct word sequence if a word is missing from the lattice. In these systems the only way to prevent the problem is to increase the lattice density to ensure the correct words are nearly always generated [LjoljeEtAl95]. This increases the computational cost of the final pass search.

The method proposed here offers an advantage because it is possible to reason about the syllable data and *re-introduce* missing hypotheses. The signal can be rescored with new segmental parameters before word hypotheses are generated. If this is well done, a better WER-density figure of merit can be expected.

9.5.1 Heuristic and Reasoning Methods

Subword phonetic or syllable data provide an opportunity for reasoning about the signal in a way that is not possible when the only information is a set of word candidates. Heuristic rules can be employed to hypothesize word sequences that would have been ruled out by the combination of acoustic knowledge with task-dependent n-grams.

In the following sentence fragment (from **ATIS2**)

```
...(pause) as early in as (uh) in the morning as
```

the hesitation word "uh" occurs as a gesture signaling the speaker's intention to restart a flawed locution. This kind of spontaneous speech phenomenon will stymie a word network algorithm in two ways: the garbage word may be poorly modeled or confused with a real word, and the sequence of ungrammatical words will be pruned by low word-pair probabilities.

Two simple heuristic rules were conceived for application to the syllable hypothesis set:

1. A sequence described as

$$\ldots syll_i, syll_{i+1}, [uh|um], pause, syll_i$$

occurring within an 1-2 second window is tagged as a possible restart event. Words extending through $syll_i$ are re-inserted into this segment of the word lattice produced by the syllable-to-word algorithm.

2. Similarly, the sequence $syll_i, pause, syll_i$ occurring within a specific window is tagged as a hesitation. The same for repeated syllable pairs within a larger window.

These rules can be tested experimentally by means of a simple cache scheme in which syllables and syllable pairs are saved within the prescribed window.

9.5.2 Rescoring the Syllable Segments with New Segmental Features

By reasoning about PSS duration and parameter evolution, non-speech events and hesitations can be detected and used to filter and drive word hypothesis generation. Proposed segmental features include: duration, spectral slopes in frequency and in time, a description of a curve in the space of Mel coefficients [Thomson95], and spectral peaks in the maxima of energy. The spectral features introduced in section 8.4.2 would also be useful. High Ω_{31} is strong evidence for fricatives. An energy peak/valley pattern indicates the vowel termination within a pseudo-syllable, and a long duration segment without this pattern is likely a non-speech event. Syllable hypotheses within such segments are safely removed from the lattice.

9.6 OOV Event Detection

Another potential application of the syllable method is for detection of out-of-vocabulary (OOV) events. The idea is to perform two searches in parallel: the first, a standard LM approach using a well-modeled subvocabulary and n-grams, and the second, a syllable grammar. Words are annotated with the likelihood $\alpha(t_{end}) - \alpha(t_{start})$ produced by the forward algorithm, and this

value is compared with the likelihood of the maximum likelihood syllable sequence, appropriately weighted to compensate for the differently constrained search space. If the syllable sequence probability exceeds that of the competing word hypothesis by a value exceeding a certain threshold, this is marked as an OOV event.

9.7 Discussion

The difficulties inherent to large vocabulary ASR systems are slowly being overcome by the application of many knowledge sources in parallel. The syllable model offers several advantages to system builders trying to integrate disparate knowledge sources. It is general enough to model a vocabulary in a domain-independent way. It provides enough contextual data to allow sophisticated reasoning and filtering algorithms to attack various difficult sub-problems. And it can be used to develop additional knowledge about pronunciation variability in a data-driven way. The phonotactic models can be used to hypothesize multiple pronunciations for frequently misrecognized words. This would constitute an automatic method for development of optimized lexical dictionaries.

The use of syllable context is being explored by a number of researchers, who have found more evidence that the syllable approach is promising. In [CookRobinson98] a neural network is used to detect syllable onsets with 92% effectiveness. This information is used to preselect one of two sets of phone models for the decoder to employ: an ordinary context independent phone model, or a *syllable-onset* version of the model. The introduction of this test decreased the word error rate 8.6% on the **Broadcast News** corpus. At ICSI (the International Computer Science Institute) a hybrid HMM/neural-net recognition system was designed with syllable recognition units, and compared with a baseline system using phoneme units. Although the syllable system was less accurate, a combined system that merged N-best results derived from each was able to achieve a 19% relative error reduction over the baseline recognizer [WuEtAl98].

Chapter 10

Design of a Fast Search Engine

It is now possible to perform, within certain parameters, large vocabulary speech recognition in real-time on a personal computer, and achieve an acceptable word error rate. It still remains a challenge, however, to make a speech engine as efficient as possible, in terms of both computation time and memory usage. If anything efficiency has become more relevant than ever. In the past the goal of research was to demonstrate that the computer speech recognition was a solvable problem. If it could be achieved in the laboratory at N times real-time, then real-time would become possible when N times faster hardware arrived, even without further algorithm improvement.

Today speech recognition is a commercial technology finding its way into the marketplace. In the future many, perhaps most, applications may be in common appliances and consumer devices. Cost will become the important factor determining whether a product is sold with it or without it. This makes it critical to build a speech engine that can add value to an existing product without the extra cost of a fast microprocessor or expensive memory chips. A speech interface that can be implemented using an inexpensive DSP board confers a commercial advantage on the manufacturer who licenses or owns it. Not surprisingly, detailed implementation descriptions for speech engines do not often appear in the technical literature.

This chapter is a sort of "how-to" for building a fast and compact decoder for a simple one-pass recognition task. The requirements of the front-end module are omitted. (The front-end computes a small vector of features, frame-synchronously, in better than real time. In some of the experiments discussed in this thesis the vector consists of 18 PLP-based floating point coefficients (section 5.4.5); in others 38 MFCC coefficients (section 5.4.6). The feature vector computation has a cost roughly on the order of computing an FFT. After the vector is handed off to the decoding module, the front-end can reuse its data structures. Neither the computational burden nor memory usage associated with the front-end is significant compared to the requirements of the decoder.)

Beginning with a general scheme based on the algorithms of chapter 6, we trade off generality for efficiency, removing data structures and computa-

tional steps that are unnecessary, and design a system tailored to the search domain which employs a minimal amount of storage and wastes very few CPU clock cycles. At the end of the chapter a figure illustrates the absolute improvement achieved when the two schemes are implemented and verified.

10.1 A General Speech Engine

Figures 10.1 and 10.2 present the data structures required in a speech engine under the assumptions of

- continuous mixture density models, using Gaussian probability density functions, which are tied to the HMM transitions (not the states),
- a general HMM representation supporting transitions between any two states, including *backward* or *skip* transitions,
- a *directed graph*, or finite state network language representation, supporting transitions between any two of the nodes, which represent speech units (HMMs),
- a recognition grammar complex enough to require beam search,
- and a statically embedded language model, with probabilities attached to arcs of the graph.

This language representation allows cycles, and therefore can generate sequences of speech units of unbounded length. In practice, either the vocabulary size must be small or the language model must be simple. A bigram language model for n words would require $O(n^2)$, transitions; trigrams would require $O(n^3)$ transitions, and so on. In this sense the first decoder described here is *not* general. It could not be used for large vocabularies in continuous speech without adding efficient mechanisms for the application of long range contexts at the levels of both the acoustic model and the language model. It also does not provide for the modeling of acoustic and word contexts that were not seen in the training data, e.g. unseen triphones.

It is nevertheless a straightforward implementation of a Viterbi decoder that could be used for isolated word recognition of any vocabulary size, or continuous speech recognition with limited vocabulary size, i.e. a lexicon that can be processed in one pass of recognition in real time. This framework is chosen for its simplicity to illustrate the development of a truly efficient decoder. At the end of the discussion we will reintroduce the issues of large vocabulary continuous speech recognition, and indicate how they can be dealt with using the very fast and small core speech engine we will develop in this chapter as a basis for a more complex system.

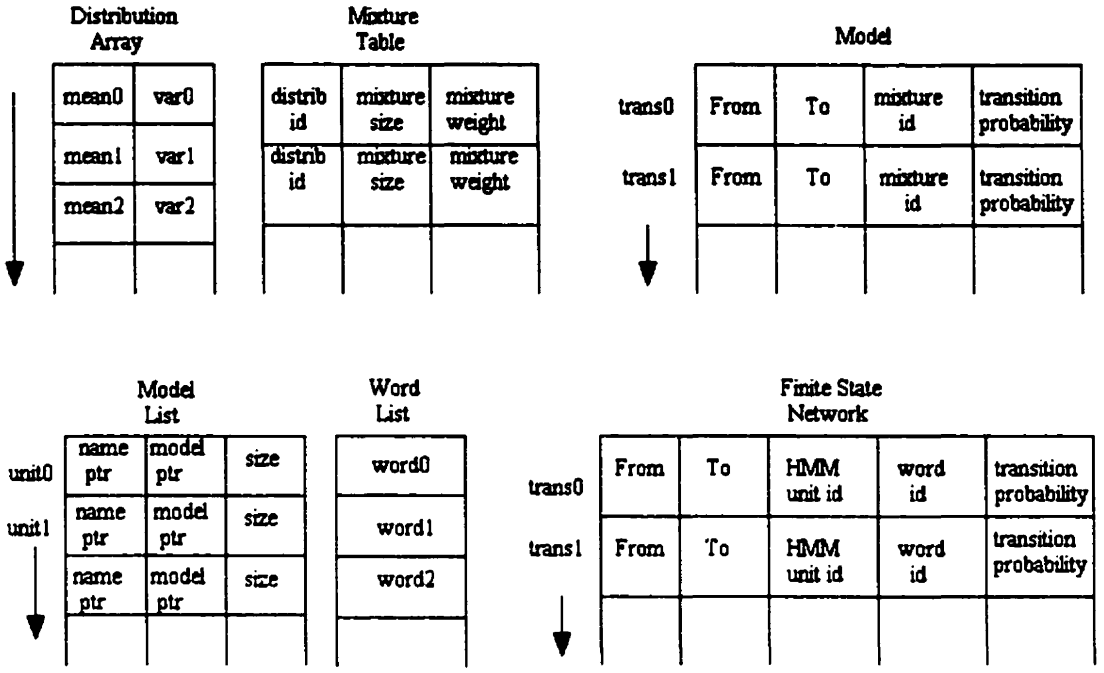

Figure 10.1: The decoder's static data structures.

10.1.1 Static Data Structures

The data structures in figure 10.1 are static, or permanent. They are used to describe the acoustic and language models.

The first two structures represent the continuous Gaussian mixture densities. The number of Gaussian components to a multivariate probability density function (pdf) is fixed, since a vector with the same dimension (i.e. the same number of features) is computed for each frame of speech by the front-end. Each pdf has d Gaussians, where d is the dimension of the feature vector. For each Gaussian there is a mean value and a variance, stored as floating point values, and these are stored in the first data structure, the *distribution array*. Each mixture contains s densities, where s can vary from model-state to model-state. This provides for different size mixtures. The components of a mixture are stored contiguously, so that the mixture can be represented with a starting index and an ending index into this array. The second structure is the *mixture table*. Each mixture is represented by three values: the start index pointing to the distribution array, the mixture size, and the mixture weight.

The third structure, the *model*, represents an HMM as an array of transitions, and contains the source state *from*, the destination state *to*, the transition probability, and an index to the mixture table.

The fourth and fifth structures are the *model list* and the *word list* (i.e.

dictionary). The *size* of each hidden Markov model, in number of states, is a one-byte value stored with each entry in the *model list*. It is still realistic to assume that there are no more than $64k$ (about $65,000$) HMM units and $64k$ words in the dictionary. This allows the other structures to use two-byte values to identify the models and words.

Assume for convenience of memory estimation that all words in the dictionary are stored in arrays of 32 bytes or less. (Of course, a prefix tree representation of the dictionary would use memory more efficiently.) Similarly, the HMM unit names pointed to by *name-ptr* in the *model list* are ascii strings stored in arrays of not more than 32 bytes.

The sixth structure is the *finite state network*, and represents the recognition grammar and language model. It is an array of transitions corresponding to arcs of the graph. This array contains the source node *from*, the destination node *to*, and the transition probability. In the case of an intra-word transition the probability is 1; an inter-word transition has a probability taken from the language model. A transition entry also contains an index to the *model list*, and an index to the *word list*. The word index defaults to -1 if the HMM not a word-ending unit.

This set of structures is one of many ways the static data can be described. Pointers can be replaced with array indices and vice-versa, depending on how substructures are allocated. HMM transition probabilities and the corresponding mixture identifiers can be stored in a matrix, or separate matrices, referenced by HMM state. This would eliminate the need for *from* and *to* fields, but for typical feed-forward acoustic HMM topologies the matrix representation would be sparse.

Let g be the number of Gaussian distributions, m be the number of mixtures, h be the number of HMM units, w be the number of words, t_h be the number of transitions per HMM, and t_n be the number of network transitions. We have already restricted the number of HMM states to less than 256, so that a byte is sufficient for the *From* and *To* values in the *model* structure. A similar restriction can be imposed on mixture size. It follows that the data structures described above would require

$$8g + 9m + (10t_h + 9 + 32)h + 32w + 16t_n \qquad (10.1)$$

bytes of memory. Note that $m <= g$ and in most cases $t_h << 256$. The memory cost is therefore of order $O(g+h+t_n)$, i.e. it depends on the number of distributions, HMM models, and network transitions. In large vocabulary systems, the number of distributions is constrained with clustered training algorithms to limit the amount of computation that must be performed at run-time. In these systems, the network size tends to have the dominant memory requirement, followed by the model space, with the distributions having the least impact.

10.1.2 Dynamic Data Structures

The data structures in figure 10.2 are used for run-time computation. They are used to explore the search space as each frame of a given signal is handed off to the decoder. This process can be viewed as expanding the next column of the Viterbi trellis (fig. 6.2). The dynamic data structures are reinitialized (or reallocated) after each frame, each block of frames, or each signal is processed, depending on the structure.

The first structure, the *TColumn*, is a place holder for the maximum incoming and outgoing likelihood for each node of the search grammar. The *input* field is used to initialize the first state of each HMM that begins at a given node of the network. This value will correspond to the maximum likelihood hypothesis that reached this node in the previous frame. The *output* field is used to store the maximum likelihood among all HMM final states connected to this node. This value will correspond to the maximum likelihood hypothesis to reach the node in the current frame.

The second structure is the *Hypothesis* array. This structure contains a list of active hypotheses (recall that we assume beam search). Each active hypothesis corresponds to an HMM unit somewhere in the network in which at least one state survived the beam applied in the previous frame. An HMM may be simultaneously active in many different places in the network. For every one of these there is an entry in the array, representing a different sequence of models, or hypothesis, that has survived the pruning process this far during the forward pass of the Viterbi algorithm.

The *Hypothesis* structure actually contains pointers to two parallel arrays: an *in* array and an *out* array which are used, much like *TColumn*, for carrying values between consecutive frames. The arrays are indexed by network transition IDs (equivalent to the indices for these transitions in the static *Network* array.) Each entry of the *in/out* arrays contains space for a pointer. When the HMM for the network transition is made active, a subarray is allocated, and its pointer is stored in the corresponding entry in the Hypothesis *in* array. A matching substructure is allocated for the *out* array. Otherwise the pointers are NULL. The subarray is indexed by HMM-state number. It contains an accumulated likelihood place holder *prob* and an accumulated time value *elapsed* for each state of the HMM.

The frame computation is as follows. For each active model-state, the *in* probability is added to the emission probability for each outgoing HMM-transition, and the result is stored in the *out* probability for the destination state of the HMM-transition. (Addition is performed because we are computing in the *log*-likelihood domain.) It is stored there only if its value is greater than the value that is there already. At the end of the frame computation for this unit the maximum likelihood among incoming HMM-transitions to each destination state is stored there.

Elapsed is the total duration, in frames, of the hypothesis. Since one frame has been consumed by the model, when the likelihood is carried for-

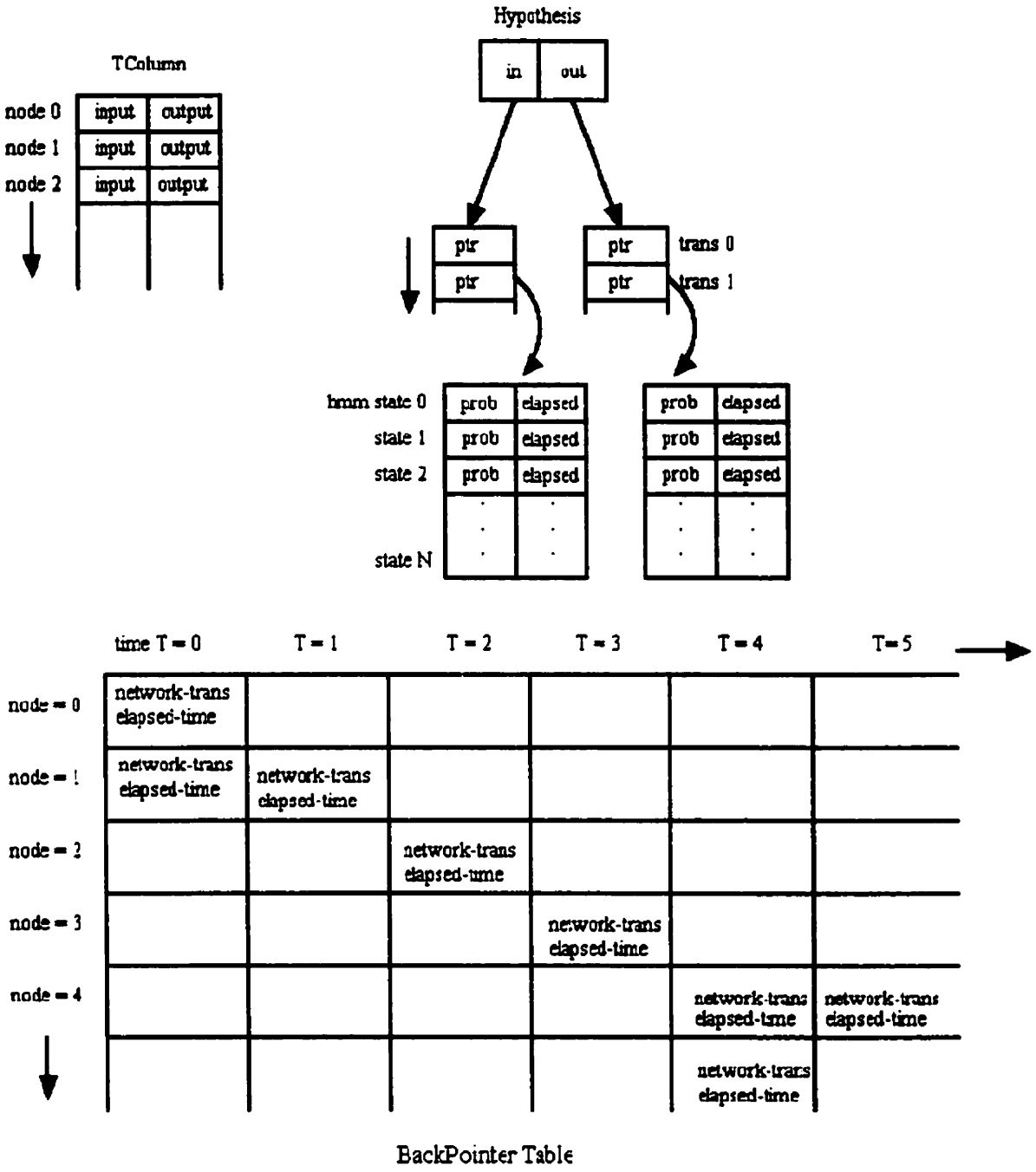

Figure 10.2: The decoder's dynamic data structures.

ward, the value of *elapsed* is incremented by 1, and carried forward as well. The likelihood and elapsed values computed in the previous frame are inputs to the current frame computation. The values computed in the current frame are stored in the *Out* substructure for the next frame. The "classical" implementation swaps the *Out* structure with the *In* structure at the end of the current frame computation, in order to prepare the new inputs. Then the new *Out* structure is reinitialized. (Actually, only the probabilities must be zeroed out.)

The last structure, *BackPointer Trellis*, stores the trellis information needed to trace the winning path backwards when the last frame, either of the signal or a block of frames, has been computed. These data must be preserved until the last frame computation. They consist of a network transition identifier and a value *elapsed-time* for each winning hypothesis at each network node in each frame. During the back-track procedure, the *BackPointer* structure is dereferenced by frame-time and by network-node. *Network-trans* identifies the preceding HMM unit in the winning sequence, and the network-node the unit came from. *Elapsed-time* indicates the time the sequence entered that HMM. When the back-track uncovers frame-time $T = 0$, the entire sequence has been recovered.

Let f be the maximum number of frames, t_n be the number of network transitions, N_n be the number of network nodes, s be the number of states per HMM, and A be the average number of hypotheses to survive the beam each frame. As before, we restrict the number of HMM states to be a byte value. The frame counts can be reasonably restricted to a two-byte value. The reusable data structures then require

$$8N_n + (8 + 4t_n) + 12sA + 6N_nf \tag{10.2}$$

bytes of memory. Note that by dynamically allocating the HMM computation subarrays as needed, we make the third term of sum 10.2 dependent on the beam width, instead of the total network size.

For any graph it is the case that $O(N_n) <= O(t_n)$. For smaller networks a large beam will be used so that $A \sim t_n$. In large vocabulary search, we use a beam such that $A << t_n$. In either case the dynamic memory cost is of order $O(t_nf)$, i.e. it depends on network size (measured in total number of transitions) and maximum signal (or block) length. As vocabulary size increases the memory cost of the network representation becomes prohibitively large, indicating where much of the optimization will come from in the next section.

10.1.3 The Search Algorithm

The following pseudo-code will perform a Viterbi beam search using the data structures described above:

```
/* initialize */
FRAME <-- 0
TCOLUMN.IN[0] <-- 1
HYPOTHESIS.IN[0][0].ELAPSED <-- 0

/* process frames one by one */
repeat
   for each active transition T in NETWORK
      HYPOTHESIS.IN[T][0].PROB <-- TCOLUMN.IN[T->FROM]
      MODEL <-- NETWORK[T].UNIT
      for each transition TRANS in MODEL
         SOURCE <-- MODEL[TRANS].FROM
         DEST <-- MODEL[TRANS].TO
         CONTRIBUTION = COMPUTE_MIXTURE(MODEL, TRANS)
         TEMP <-- HYPOTHESIS.IN[T][SOURCE].PROB + CONTRIBUTION
         if TEMP is within the beam and TEMP > HYPOTHESIS.OUT[T][DEST].PROB
            HYPOTHESIS.OUT[T][DEST].PROB <-- TEMP
            HYPOTHESIS.OUT[T][DEST].ELAPSED <--
                                       HYPOTHESIS.IN[T][SOURCE].ELAPSED + 1
            if DEST is final state in MODEL and TEMP > TCOLUMN.OUT[T->TO]
                  TCOLUMN.OUT[T->TO] <-- TEMP
                  BACKPTR.OUT[FRAME][T->TO].TRANSITION <-- T
                  BACKPTR.OUT[FRAME][T->TO].ELAPSED <--
                                       HYPOTHESIS.OUT[T][DEST].ELAPSED
            endif
         endif
      endfor
   endfor
   swap TCOLUMN.IN, TCOLUMN.OUT
   swap HYPOTHESIS.IN, HYPOTHESIS.OUT
   FRAME <-- FRAME + 1
until last frame received

/* find winning node in last frame */
MAX = -MAXFLOAT
for each NODE in network
   if TCOLUMN.OUT[NODE] > MAX
      MAX <-- TCOLUMN.OUT[NODE]
      WINNER <-- NODE
   endif
endfor

/* follow back-pointers to recover recognition sequence */
NODE <-- WINNER
while FRAME > 0
   T <--  BACKPTR[FRAME][NODE].TRANSITION
   output(NETWORK[T].UNIT)
```

```
   FRAME <-- FRAME - BACKPTR[FRAME][NODE].ELAPSED
   NODE <-- NETWORK[T].FROM
endwhile
```

10.1.4 Analyzing the Computational Cost

The algorithm can be analyzed in terms of number of operations, given f frames of signal, t_h transitions per HMM, and A active hypotheses on average per frame. Details of the beam computation, while not insignificant, are omitted.

In the analysis, the following abbreviations apply:

- **ASN**: an assignment, or data store, operation
- **CMP**: a comparison operation
- **INT**: an integer math operation
- **FLT**: an floating-point math operation

The mixture density size (number of Gaussians) times the dimension of the pdf (the size of a feature vector) is dd. The first block of pseudo-code, *"initialize"*, consists of 3 **ASN** operations. The second block, *"process frames"*, consists of approximately

$$f[T_n(\mathbf{INT} + \mathbf{CMP}) + 7\mathbf{ASN} + \mathbf{INT}] + fA(5\mathbf{ASN} + 2\mathbf{CMP}) + fAt_h[4\mathbf{ASN} + 4\mathbf{CMP} + 1.5\mathbf{INT} + (3 + 4dd\mathbf{FLT})] \quad (10.3)$$

operations. The final block of pseudo-code, *"follow back-pointers"*, consists of $f(4ASN + INT)$ operations. Clearly, the computation for each frame depends on three quantities: the size of the network, the number of Gaussians, and the size of the search beam. For a small network with large mixtures, and a wide search beam, the computation is dominated by the floating point operations of the mixture density computation in the third term of approximation 10.3.

10.2 Optimizations

The decoder described in the previous section can be optimized in many ways, by trading off its generality for improvements in memory and time. Some of these tradeoffs are obvious, other less so. We will restrict the recognition problem to the following:

1. Isolated word recognition.
2. Fixed word-sequence continuous speech recognition. Examples: spelled name recognition for call routing; phrase recognition.

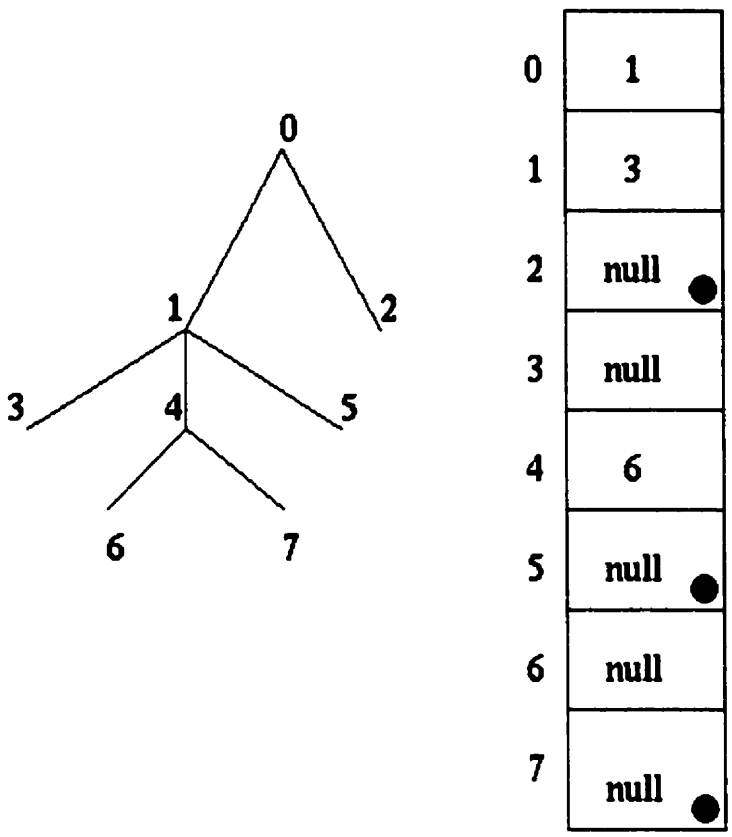

Figure 10.3: A minimal tree representation for the network.

These kinds of tasks can have a different network representation: trees instead of graphs. From this simplification we can derive many new efficiencies. The network, as we have seen, dominates the memory space of the decoder. By eliminating cycles we can use a minimal tree representation for the network (figure 10.3), which reduces the memory requirements of the static data structure, and eliminates the need for some of the dynamic structures.

Figure 10.3 describes the topology of the illustrated tree with a compact array containing a breadth-first representation. Each slot in the array represents a node N_i. It contains the array index of the first child node attached to N_i. The remaining child nodes are those array slots that sequentially follow the first child. The last child of N_i is denoted by the setting the most significant bit of the value contained in the slot. (In the figure the set bit is illustrated by a black dot.) If a node has no children, the corresponding array slot contains a distinguishing special value (denoted *null* in the figure), which functions as a null pointer.

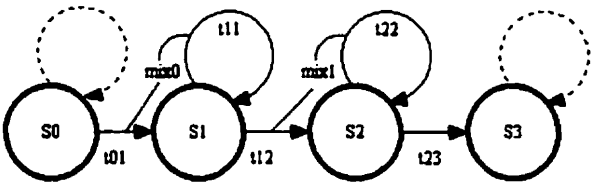

Figure 10.4: A simple feed-forward HMM topology.

A tree contains a single path from leaf to root. This means each leaf corresponds to, and *identifies* exactly one speech sequence. No back-track is required to recover the recognized string, only a pointer from the leaf node to the lexicon of the decoder. Eliminating the back-track procedure removes the need for the expensive *BackPointer* structure, which as seen in sum 10.2, requires an allocation of memory proportional to network size *and* signal length.

Another restriction we introduce is a restriction of HMM topologies to a simple feed-forward structure, as in figure 10.4. The HMM topology is completely described by the number of states. The transitions are implicit. Every state emits a self-transition and a forward transition. The same mixture is tied to the incoming transition *and* the self-transition of each state. In the figure, there is a dotted self-transition attached to the first and final states. This is because these two implied transitions are not computed; in practice they do not exist. Similarly, there is no mixture tied to the incoming transition to the final state, because this transition is *non-emitting*, and does not consume a frame.

The loss of generality is less serious than it might seem. In practice these simple topologies are the ones most often used for acoustic modeling, and as seen in chapter 8, the improvements gained by deviating from them are small. As we shall see, the added efficiency gained from a regular topology speeds up search enough to more than offset whatever might be lost in accuracy.

10.2.1 Static Data Structures

The computational inefficiency of the network and HMM representation in section 10.1 derives in large part from the need to endlessly traverse the language and model graph structures during the decoding process. Simplification and regularization of these structures eliminates much of this repetitive lookup process, and it eliminates extra data elements that support the lookup process.

The simplified static data structures are illustrated in figure 10.5. The *Distribution*, and *Mixture* tables are unchanged. The Network structure no longer needs to store the *From* data, since the tree is always traversed downwards. The *Model* structure does not need *From* or *To* information. The

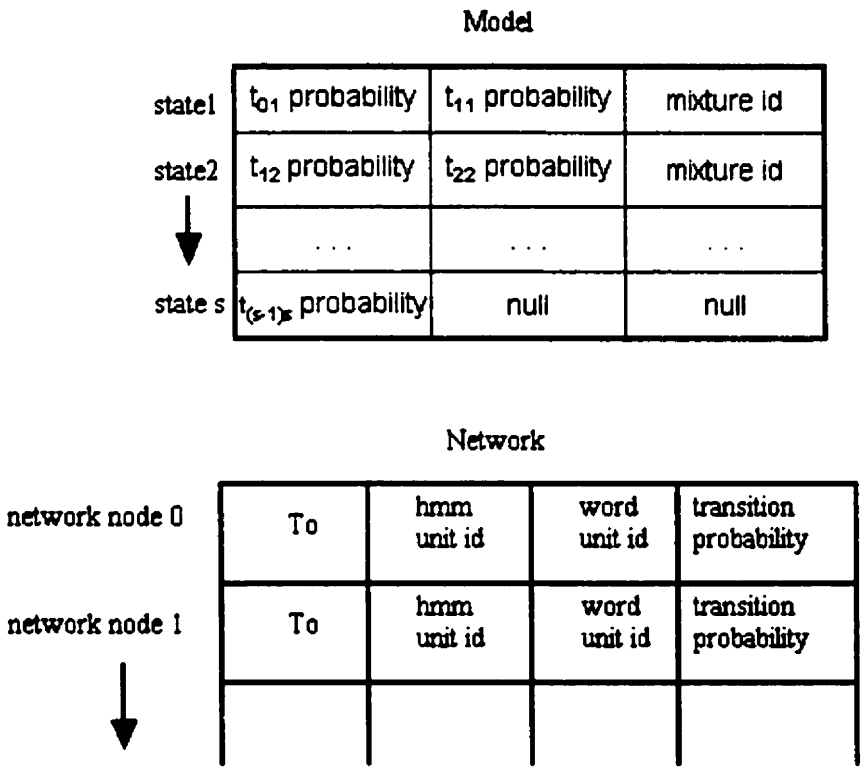

Figure 10.5: Simplified static data structures.

topology is completely defined by the number of states which is stored in the one-byte variable *size* in the *Model List* (figure 10.1).

Let g be the number of Gaussian distributions, m be the number of mixtures, h be the number of HMM models, s_h be the number of states per HMM, and t_n be the number of network transitions. It follows that the static data structures now require

$$8g + 9m + (12s_h + 9 + 32)h + 32w + 12t_n \tag{10.4}$$

bytes of memory. Compare sum 10.4 to 10.1. The network cost is reduced 25%. Since s_h is roughly half the number of transitions for a given unit's topology, the memory cost of the HMM representations is reduced as well. Even greater space saving is accomplished with respect to the dynamic structures.

10.2.2 Data Structures for Simplified Search

The modifications to the dynamic data structures are motivated by a simple objective: we wish to store the dynamic data with the hypotheses that accumulate during search. In doing so we will allocate an amount of search-related memory directly proportional to the width of the search beam (i.e. the number of active hypotheses) and *not* to the size of the language-network model.

A second advantage emerges from the restrictions on the network and model topologies. Because there are no backward transitions, it is possible to compute the likelihoods of network-nodes and model-states in order and *in place*. There is no need for an *In* and *Out* place holder, and no need for these structures to be swapped at the end of a frame computation. In contrast, when computing likelihoods over a graph with cycles, you must preserve all the input likelihoods until the end of the frame computation. As long as there are still transitions left to process, some of these may depend on previously encountered input values.

The dynamic data structures are modified in the following ways:

- There is no *TColumn* structure. An input value for the next column of the trellis is stored with the *Hypothesis* data that depend on it.
- There is no *BackPointer* structure. When a leaf node is reached, it points directly to the recognized sequence.
- Each hypothesis is stored with just one subarray which holds the state likelihoods of the corresponding HMM unit. This subarray is processed in reverse order, from the last state to the first, so that an input state likelihood is overwritten only after the destination state is computed.
- A hypothesis is stored with an identifier corresponding to a node in the *Network* array. This node is a destination node of a transition in the network tree.

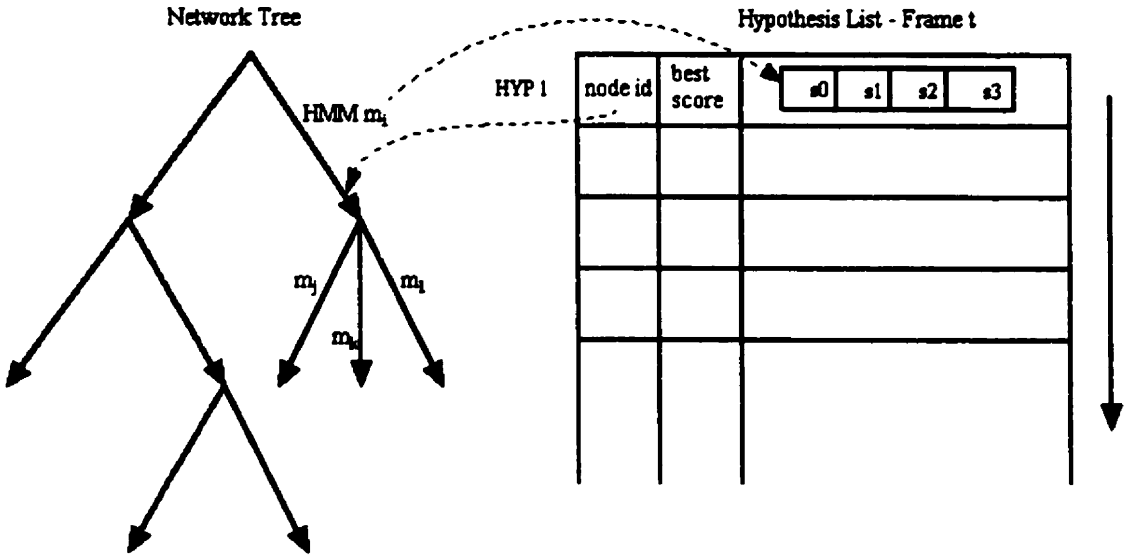

Figure 10.6: The new search algorithm is driven by the active *Hypothesis* structure.

Since the network is a tree, each node has exactly one input transition. Thus the destination node is sufficient to denote the transition. The search algorithm references this node in *Network* in order to identify the HMM unit that feeds into it, and the child nodes connected to it. This lookup step is the only time during the forward processing that the network array is checked.

The new *Hypothesis* structure is illustrated in figure 10.6. It contains fields for network *node* identifier, and the input likelihood values for each state of the corresponding model. State 0 stores the incoming likelihood, computed in the previous frame, for the network node. There is also a value *bestscore* used for the beam threshold computation. The data structure requires

$$(8 + 4s)A \tag{10.5}$$

bytes of memory, where s is the number of HMM states and A is the average number of active hypotheses. Contrast this with the memory estimate 10.2.

10.2.3 The Search Algorithm

The search algorithm works by processing a list of hypotheses, allocating the memory space for the dynamic data within this list, and reusing the the allocated space for a hypothesized network transition from one frame to the next. Both time and memory are directly proportional to the quantity of active hypotheses. That is why the resulting decoder is much more efficient.

The algorithm processes the list of active hypotheses twice. The first pass computes the new output likelihoods, and finds the best one for the purpose

of computing a beam threshold. The second pass compares each final-state likelihood to the beam threshold. If it falls within the beam, the network structure is used to determine which units are connected to the output of this hypothesis model. Each of these is added to the list of hypotheses, if it is not there already. Then the likelihood is copied to the initial-state placeholder for each of the connected units. The second pass also takes care of removing hypotheses from the active list, as follows. The best non-final state likelihood for a hypothesis model is compared to the beam threshold, and if it does not lie within the beam, then this hypothesis is removed from the list. If it does lie within the beam, then this hypothesis remains in the list so it can be processed in the following frame.

The pseudo-code for the new search procedure is as follows:

```
/* initialize */
NUM_HYPOTHESES <-- 1
HYPOTHESIS[0].NODE <-- 1
HYPOTHESIS[0].PROB[0] <-- 1

/* process frames one by one */
repeat
   for each hypothesis I
      HYPOTHESIS[I].BESTSCORE <-- -MAXFLOAT
      MODEL <-- NETWORK[HYPOTHESIS[I].NODE].UNIT
      for each state STATE in MODEL from last to second
         CONTRIB <-- COMPUTE_MIXTURE(MODEL, STATE)
         R1 <-- HYPOTHESIS[I].PROB[STATE-1]+MODEL[STATE].IN_TRANS_PROB+CONTRIB
         R2 <-- HYPOTHESIS[I].PROB[STATE]+MODEL[STATE].SELF_TRANS_PROB+CONTRIB
         TEMP <-- MAX(R1, R2)
         HYPOTHESIS[I].PROB[STATE] <-- TEMP
         /* The following keeps a maximum likelihood over the states
            of this hypothesis model. All though not shown here, an
            optimal value over all states computed for the frame is
            also preserved */
         HYPOTHESIS[I].BESTSCORE <-- MAX(HYPOTHESIS[I].BESTSCORE, TEMP)
      endfor
   endfor
   for each hypothesis I
      if HYPOTHESIS[I].BESTSCORE is not within the beam
         REMOVE(I)
      else
         NODE <-- HYPOTHESIS[I].NODE
         LAST <-- index of last state of NETWORK[NODE].UNIT
         if HYPOTHESIS[I].PROB[LAST] is within the beam
            for each node CHILD in NETWORK[NODE].TO linked list
               if CHILD is active in some hypothesis J
                  HYPOTHESIS[J].PROB[0] <-- HYPOTHESIS[I].PROB[LAST]
               else
```

```
                NEWHYP <-- ADD(CHILD)
                HYPOTHESIS[NEWHYP].PROB[0] <-- HYPOTHESIS[I].PROB[LAST]
             endif
          endfor
       endif
    endif
 endfor
until last frame received

/* find winning node in last frame */
MAX = -MAXFLOAT
for each hypothesis I in list
    NODE <-- HYPOTHESIS[I].NODE
    LAST <-- index of last state in NETWORK[NODE].UNIT
    if HYPOTHESIS[NODE].PROB[LAST] > MAX
       MAX <-- HYPOTHESIS[NODE].PROB[LAST]
       WINNER <-- NODE
    endif
 endif
endfor

/* output answer */
output NETWORK[WINNER].WORD
```

The *ADD()* and *REMOVE()* list modifying functions should employ binary lookup of an ordered hypothesis list, to achieve an $O(A \log A)$ rather than an $O(A^2)$ computational overhead. This can be achieved by use of a *red-black* tree or a *B*-tree structure for the *Hypothesis* data. This adds 8 bytes of pointer data for each *Hypothesis* entry. We tolerate these space and time costs when $A << N$, where N is the size of the *Network*. If this assumption does not hold for a particular recognition task, we can trade off memory in the *Network* structure for more efficient (linear-time) processing of the active *Hypothesis* list: a pointer can be store with each network node to the corresponding *Hypothesis* entry. In this case insertions and update operations would be applied by adding the appropriate hypotheses to a second list, a simple array to be processed in the following frame. Hypotheses pruned from the active list would be simply deallocated.

10.2.4 Analyzing the Computational Cost

In contrast to the general decoder sketched out in the first part of this chapter, the new algorithm has no fixed network cost. All three blocks of pseudocode have either a constant number of operations or a computational cost dependent on the number of active hypotheses. As before, the algorithm can be analyzed in terms of number of operations. Let f be the number of frames of signal, s_h the number of states per HMM, and A the active hypotheses on

Problem Size	Decoder Version	Memory Usage	Speed (x Real-Time)
3,000 words	General	12,860k	0.12
3,000 words	Tree-based	1,152k	0.067
30,000 words	General	210M	0.97
30,000 words	Tree-based	8,708k	0.12

Table 10.1: Time/memory comparison of general and optimized decoders.

average per frame. The mixture density size times the number of coefficients is dd, and the average branching factor of the network tree is bf.

The first block of pseudo-code, *"initialize"*, consists of 3 **ASN** operations. The second block, *"process frames"*, consists of approximately

$$fA[2\mathbf{ASN} + 2\mathbf{FLT} + 2\mathbf{CMP} + s_h(10\mathbf{FLT} + 5\mathbf{ASN} + 2\mathbf{CMP} + 4dd\mathbf{FLT}) + bf(\mathbf{INT} + 2\mathbf{CMP} + 2\mathbf{ASN} + 2(\log A)(\mathbf{INT} + \mathbf{CMP} + \mathbf{ASN}))]$$

operations. The final block of pseudo-code, which retrieves the winning node, consists of $A(4ASN + FLT + CMP)$ operations. Clearly, the computation for each frame depends strongly on only two quantities: the number of Gaussians, and the size of the search beam. The ordering of the hypotheses in a list necessitates $O(\log A)$ operations to find, add or remove them. If we disregard constants we find the algorithm has complexity $O(A \log A)$; i.e. it depends entirely on the search beam width and not on the size of the language.

The efficiency of this algorithm compared to the initial, general implementation is made clear in table 10.1. The two decoders were tested on grammars of up to 30,000 spelled street names. The tests ran on a Pentium-II workstation, with a 450 mHz clock and 256 megabytes of memory. The test data consisted of 160.1 seconds of speech. For the large test (30,000 names) the search algorithm achieved a recognition accuracy of 94.6%. The optimized engine performed the same search in one eighth the time, while reducing memory requirements from 210 to less than 9 megabytes.

10.2.5 Enhancements for Continuous Speech

The stripped down speech engine of this chapter is easily modified for continuous recognition tasks, while preserving most of its inherent efficiencies. The case of continuously spoken utterances from a language of fixed sequences is exemplified by spelled word recognition. This language can be implemented as a tree with one modification: each node of the tree has an implicit looped *silence* model between it and its successors. The regularity of this language allows us to compute it without the considerable overhead of inserting all these extra nodes in the *Network* structure. Whenever a node enters the active hypothesis list, a corresponding looped silence model is added to the list

as well. The identity of the silence hypothesis is unique in (Network Node id, Unit id).

For general language recognition, we restore the data and computational overhead associated with the *BackPointer* structure. What is required is a *lattice* of hypothesized word endings in signal-time from which to trace the best or N-best sequences. Also the initializing of hypothesis model-state zero is more complicated; the very best incoming likelihood and its history must be selected among all incoming transitions. But we continue to benefit from the efficiencies of the simple HMM topology, the *in place* computation of state likelihoods, and the elimination of costly, repetitive processing of network transition arrays and model transition arrays.

Finally, the reader should note that very large vocabulary continuous speech recognition requires more sophisticated recognition algorithms, often based on multiple passes of search. Each such pass will require its own set of optimizations. However, a form of the decoder described here can serve as an efficient first-pass search engine for a fast match, generating a phone lattice or word graph from subword HMMs.

Chapter 11

Spelled Letters: Algorithms and Application

This chapter is about robustness improvements that were made to a real, working continuous speech recognition system. A speaker-independent speech recognizer for continuously spelled names, implemented for a switchboard call-routing task, was analyzed for sources of error. [1] The results of a field trial indicated that most errors were due to (1) extraneous speech, and (2) end-point detection errors.

To attack the second problem, a strategy is proposed for improving the system tolerance for speech with pauses. To deal with the first, and more intractable problem, a new algorithm is introduced for spotting spelled-word sequences in a signal containing extraneous speech. Experimental results in the laboratory show that with the letter-spotting algorithm, the name retrieval error-rate is reduced by 60.7% on signals with extraneous speech, from an absolute error rate of 75.8% to 29.8%. On clean speech the absolute error rate increases from 4.5% to 5.5%. On data collected during a follow-up trial of the working system, name retrieval error decreases by 54.1% from 23.3% to 10.7%, an improvement solely due to the new letter-spotting algorithm. [2]

11.1 Introduction

The initial system studied in this chapter is a multi-pass HMM-based decoder for spelled-name recognition over the telephone, described in [JunquaEtAl95] (see also [Junqua97]). This system is called $SmarTspelL^{TM}$. The first pass

[1] The system served as a telephone autoattendant at Speech Technology Laboratory - Panasonic Technologies Limited, in Santa Barbara, California. It was active from about 1994 to 1997. By 1998 it had been replaced by a whole-name recognizer. In the new system spelled name recognition served as a fall-back procedure.

[2] This algorithm is the subject of a patent filing entitled "Speech Recognition System Employing Multiple Grammar Networks," by Michael Galler and Jean-Claude Junqua, filed April 16, 1997 (H07-1222) in the U.S., and filed in Japan, Taiwan and China (H07-1222).

employs a bigram language model and a Viterbi beam search to produce the N-best (typically 20) hypothesized letter sequences. In the second pass, the sequences are compared to a dictionary and a dynamic-programming match is used to select the N-best names in the name directory, based on statistics for letter-confusion. A third pass of recognition is then performed in which the acoustic parameters of the signal are fitted to the HMM sequences representing the N-best names from the directory. In this final pass a full Viterbi search is performed on the reduced set of candidate names. In this way an accurate, efficient search is produced by filtering the search space with different constraints through various stages.

The recognizer was integrated with the telephone switch at a corporate business office. [3] Several months of field tests were logged in which callers used the system to direct their calls to company staff members. Callers were asked to confirm the correctness of the recognition output with a yes/no response. Calls were recorded, logged, and later transcribed to determine the source of errors.

Figure 11.1 presents the collected performance statistics. According to the experimental data, the two most common sources of error were pauses interposed in the letter sequence, causing a premature detection of end-of-speech, and extraneous speech, usually at the beginning, e.g. "Smith [pause] s-m-i-t-h" Other major sources of error include low signal energy collected through speaker-phones, and other effects of channel mismatch between training and test conditions.

11.2 Related Work

11.2.1 Call Routing

The work described here involves a number of different issues, including call-routing by name recognition, spelled-word recognition, word-spotting techniques, rejection of noise and extraneous speech, and the design of user interfaces for telephone-based speech recognition systems. Speech-operated auto-attendants are now being deployed by many institutions. A number of systems have been introduced experimentally for call-routing that involve speaker-independent, large-directory, whole-name recognition [SmithBates93], [YamamotoEtAl94], [BilliEtAl95]. Some systems have employed hybrid schemes that attempt to manage errors through dialogue, prompts, and user-feedback [JohnstonEtAl96], [KellnerEtAl96], and a mixture of whole-name and spelled-letter fallback [JohnstonEtAl96]. In contrast with these systems, [FraserEtAl96] attempts to perform call-routing for small directories on small, inexpensive

[3] The implementation of $SmarTspelL^{TM}$ together with its design as an autoattendant was the subject of U.S. Patent #5,799,065 (H07-1221), entitled "Call Routing Device Employing Continuous Speech", by Jean-Claude Junqua and Michael Galler, approved August 25,1998.

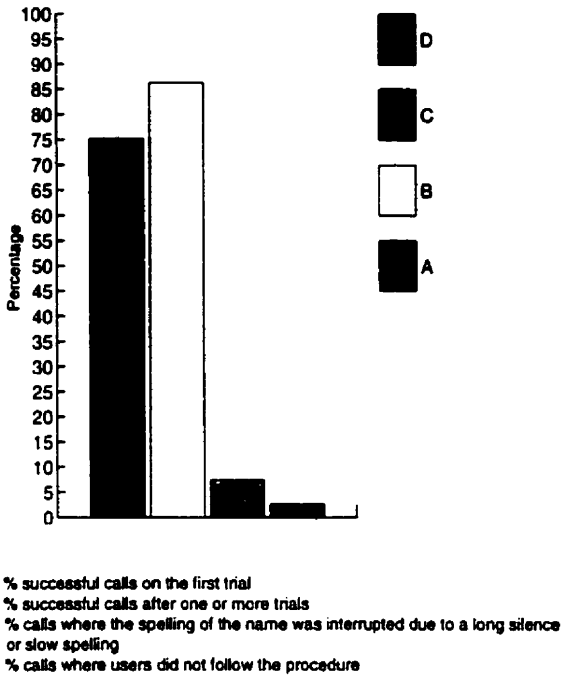

Figure 11.1: Call recognition and error rates.

hardware, based on whole-name recognition.

The system described here can be seen as a competitor to these systems because it manages name recognition using a small set of acoustic models that run quickly on simple hardware. Also, because of the extra information exploited in the lexical comparison to the dictionary in this multi-pass algorithm, the recognition accuracy is better than for conventional systems which put all the spelled names into a network [Junqua97]. It can also be seen as a complementary technology, or fall-back procedure that can be employed by the larger, whole-word systems.

11.2.2 Robustness Issues

A key robustness issue addressed here is the problem of Out-Of-Vocabulary (OOV) rejection. Although the user instructions are very simple (paraphrase): "Please spell the name of the person you want to call", users have a tendency to pronounce the whole name anyway. Well known techniques exist to deal with this problem, based on the use of filler models and multiple grammars [RoseEtAl95], [RahimEtAl95]. However, in this work the integration of those techniques with the multi-pass spelled name search is original. [4] Other techniques recently introduced for OOV rejection and keyword-

[4]The work described here has been previously published by the author, with co-author Junqua, in the 1997 ICASSP proceedings.

spotting are based on on-line garbage modeling [BourlardEtAl94]. These techniques, while not explored in this thesis, are strongly complementary to the algorithms introduced here.

11.3 Data Description

The database used in these experiments is a subset of the speech telephone corpus collected at Oregon Graduate Institute (**OGI**) [ColeEtAl91]. Over four thousand people telephoned in response to public requests. They were prompted by a recorded voice to spell their first and last names, with and without pauses, together with other information. 60 pre-labeled repetitions of the alphabet (which did not belong to the test set as defined in the **OGI** CD-ROM) and more than 1200 different calls were used for training the HMMs. 558 calls were used for the validation set, and 491 calls for the basic test suite. The purpose of the validation data was to optimally tune the system's parameters before running it on the test set. As every speaker belongs only to one set (training, validation or test) the experiments conducted are speaker-independent.

All data in these three sets were last names spoken without pauses, and none of them contained extraneous speech, line noise or speech related effects such as lip-smack or breath noises (as transcribed in the database). These three sets were subsets of the corresponding training, test and validation sets defined in the **OGI** CD-ROM. They were used in the design of the base system, whose evaluation lead to the further enhancements of this chapter.

In order to test the multiple-grammar, letter-spotting algorithm introduced here, two new data sets were assembled from the **OGI** corpus. The first consists of 521 signals containing extraneous speech, in which the pronounced name precedes the spelled name. In addition, a set of 159 spelled names containing non-speech utterances and noises was used for a laboratory evaluation of the improved system.

During the first field trial 403 calls were recorded over the telephone. Roughly half were in-house transfers using the digital PBX, and half were made from outside the company through the telephone network. These data were also used to evaluate the initial recognition system.

11.4 Enhanced Robustness by Pause-handling

Many errors were caused by the premature detection of end-of-speech by the Voice Activity Detection (VAD) algorithm. These events were triggered when the speaker hesitated either during the spelling of a name, or between some introductory utterance and the subsequent utterance of spelled letters. The VAD algorithm is implemented as a state machine with 4 states: *non-speech*, *speech-in-progress*, *end-of-speech*, and *false-alarm* (a non-speech event).

State changes are triggered by changes in signal energy, computed adaptively at run-time. Frames of speech data detected by the VAD are passed from the front-end to the recognition module, which computes the forward probabilities frame-synchronously. The backtrack phase of the Viterbi recognition process is triggered by a VAD-detected end-of-speech. In real calls valid speech segments are often interspersed with pauses which, in the original system, caused the VAD to trigger the recognition process prematurely.

The spelled-name recognition algorithm was modified to allow signals containing pauses to be recognized. In the modified system, the VAD continues to classify and segment the raw signal as before; however, the recognition module employs a timeout of its own to decide whether an input has terminated. The new algorithm allows up to two seconds of silence between letters before determining the final endpoint was reached. During this interval, it proceeds through the stages of recognition, preparing a tentative output. If the VAD determines the speaker has resumed speaking before the timeout, the second or third pass of recognition aborts and the forward algorithm of the first pass resumes until the next pause is detected. When two seconds of non-speech have elapsed, the tentative response is confirmed and delivered, and recording stops.

In the original field test [JunquaGaller96], 7.4% of calls were interrupted due to the problem of slow or interrupted speech. With the modified system described above, less than 2% of calls are subject to this kind of error.

11.5 Modeling of Extraneous Speech and Noise

11.5.1 Improved Robustness with Letter-Spotting

A modified recognition strategy was proposed for dealing with the other major source of error. Extraneous speech is managed with a word-spotting strategy (Figure 11.2.) An initial experiment employed a network with filler to produce all the N-best sequences of letters for the dictionary alignment. However, for input signals which do not contain extraneous speech this algorithm increased the instances in which the alignment procedure failed to produce the right name in its N-best list. This was caused by the increased number of unit errors present in the input sequences, due to the additional complexity of the language model.

A more successful strategy was to fit the signal to two different networks. The first network consists of a filler HMM, followed by silence, followed by the letter models. This network tries to spot the best letter sequence following an initial extraneous noise, word, or phrase. The second network assumes only letters are spoken, and tries to fit the whole signal to the best letter/pause sequence. In both cases the transitions between letter HMMS are weighted by bigram probabilities. Each of these two recognition passes produces an N-best list of sequences. In the next stage of recognition, the sequences are

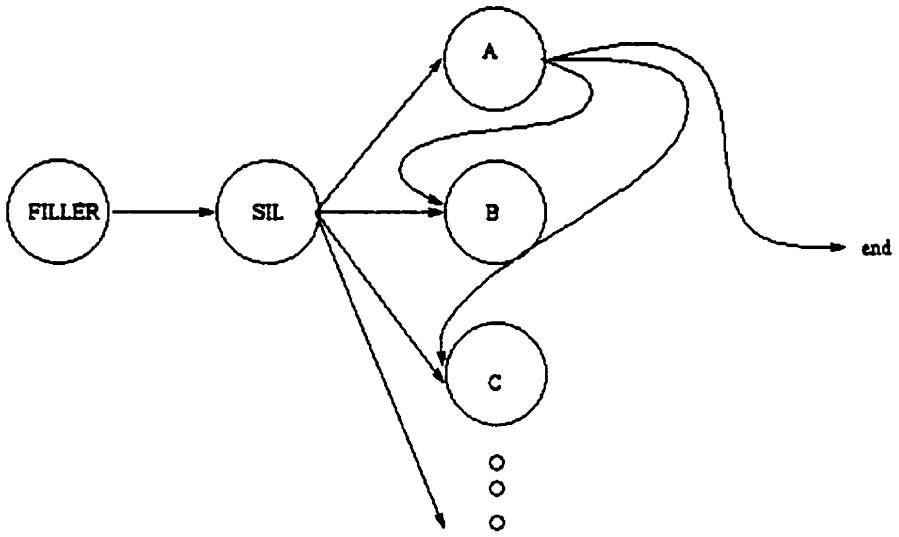

Figure 11.2: Network for letter-spotting.

aligned against the name dictionary to retrieve the most likely names with respect to the confusion statistics on letters.

Two alignments are made to compare letter sequences to the names in the dictionary. Since the network without a filler model is used, the input to one alignment procedure is guaranteed to include the best sequences that the letter models can produce for a signal not containing extraneous speech. If the signal does contain extraneous speech, the network containing filler is more likely to produce a sequence acoustically similar to the actual spelled name, and the input to the second alignment procedure will contain that sequence. In the new algorithm, the output of each alignment produces 10 hypothesized names, and these are combined into a list of 20 candidates. In the final pass a dynamic grammar is generated consisting of the best candidates from the dictionary, with an optional initial filler model. A Viterbi acoustic recognition pass, using detailed wide-beam search, is performed with this grammar to select the most likely name.

11.5.2 Filler Models and Networks

Different kinds of filler models were constructed and tested. One consisted of a single speech unit trained on all non-silent segments of the **NTIMIT** 8kHz training corpus, followed by a silence unit. The filler unit was a left-to-right model containing 8 states, with 8 Gaussian distributions per mixture density. The grammar containing this sequence segments the input utterance into an initial, extraneous burst of speech, followed by a silence, and ending with a sequence of letters. This network is motivated by analysis of the error

Data Set	Size	Original System Error	Letter-spotting System Error
names only	491	4.5%	5.5%
names with extr. speech	521	75.8%	29.8%
names with noise	159	28.3%	11.3%
field trial data	403	23.3%	10.7%

Table 11.1: Name retrieval experiments – Results.

data, in which a spelled name was often preceded by an introductory word or phrase, and then by a pause of some duration.

The second kind of filler tried was a looped-phoneme network trained on **NTIMIT**. This network modestly increased the number of correct outputs from the dictionary but at a cost of increased computation. Because of the success of the simple filler model as seen below, this approach was dropped.

11.5.3 Noise Modeling

A final variation on the filler network was tried, in which the filler models described above were replaced with a small set of noise models, lip-smack, breath-noise, and line-noise, trained on the **OGI** database. These 4-state, 16 distribution models were added to the letter-recognition network, and tested for accuracy in detection of extraneous noise or speech in the spelled-name data. The performance of the filler-based letter-spotting was compared to this method of using models trained for specific noises.

11.6 Experimental Results

The results for the original and modified name recognition algorithms are summarized in table 11.1. The results are presented as name-retrieval error, or the percentage of recordings for which the wrong name was selected at the end of the final pass.

The original system was shown to achieve a 4.5% retrieval error rate (on 491 **OGI** test signals) for a 3,388 word dictionary. When tested on a separate subset of 521 **OGI** signals consisting of extraneous-speech *plus* spelled-name data, the same system had an error rate of 75.8%. When the letter-spotting algorithm was tested on the spelled names, recognition error increased from 4.5% to 5.5%. However, on the extraneous speech data, the error rate decreased by 60.7% from 75.8% to 29.8%. On the noisy signals, there was a similar 60.1% error reduction.

Data Set	Letter-spotting with Filler Model	Letter-spotting with Noise Models
names with extr. speech	29.8%	56.0%
names with noise	11.3%	11.3%

Table 11.2: Error rates of letter-spotting by filler and by noise models.

Of particular interest was the comparison between the filler model and the noise model performance on data containing noise, as seen in table 11.2. As should be expected, the filler performed much better than the noise models at detecting extraneous speech. But the filler model performed equally well in name retrieval as the noise-modeling network in detecting speech with noise. However, better performance may be achievable with more precise noise models.

11.7 Conclusions

Through the careful examination of the performance of a speech recognition service on live data, it is possible to identify the most serious issues of robustness. Often these problems can be greatly alleviated by small adjustments to the user interface. In other cases new and better algorithms are needed. The multiple *grammar-in-parallel* approach is effective at improving the robustness of a recognition system when extraneous utterances will appear in a small but significant percentage of voice inputs to the system. It is better than a standard keyword-spotting algorithm, because it generates multiple N-best lists in parallel: one list that assumes canonical voice inputs, and one or more lists for degenerate cases assumed to contain extraneous speech or ill-formed utterances. In this way a combined set of candidates is produced which will contain the best hypotheses under *either* assumption. The final pass of search can then find the right candidate with a high probability of success. Further improvements to the spelled-name recognizer of this chapter are described in [RigazioJunquaGaller98]. The goal of this later work was to improve the discriminative power of the letter models, particular among the confusable *E-set* of the alphabet. These remain a challenge for accurate discrimination, particularly over the bandwith-limited telephone channel. Techniques of discriminative training were developed to simultaneously optimize for discriminative power the HMM state-weights for the letter models, the statistical language model on the letters, and the heuristic weighting of the language model. [5]

[5]Further improvements were the the subject of a U.S. Patent entitled "Method and Apparatus Using Probabilistic Language Model Based on Confusable Sets for Speech Recognition", by Luca Rigazio, Junqua and Galler, filed March 23, 1998 (H09-0921).

Chapter 12

Enhancements of Speech Production in Esophageal Speakers

This chapter describes an experimental medical application of speech recognition technology. Esophageal speakers are patients who, generally because of cancer, have had part or most of their larynx surgically removed. Lacking glottal tissue, esophageal speakers are trained during post-operative rehabilitation to produce a voice source by bringing about a vibration of the esophageal superior sphincter. They must fill the esophagus with an injection of air before every utterance, thus creating an air reservoir to drive the vibration. The resulting gulping noise is disconcerting for both the speakers and listeners. This chapter describes a method for the automatic recognition, and suppression by electronic means, of the injection noise which occurs in esophageal speech.

12.1 Introduction

People who have had laryngectomies can be retrained to speak with a variety of techniques, some involving prosthetic devices. The artificial larynx, a hand-held device which introduces a source vibration into the vocal tract by vibrating the throat externally, is the easiest for patients to master, but does not produce airflow. Without that flow the intelligibility of consonants is diminished. Tracheo-esophageal speech uses a prosthesis to divert outgoing lung air into the esophagus, bringing about a vibration of the esophageal superior sphincter. This method does produce airflow for consonants and permits utterances of normal duration. However, it requires a surgically produced connection between the esophagus and the trachea, and is not suitable for some patients.

Esophageal speech, which requires speakers to *insufflate*, or inject air into the esophagus [WeinbergBosna70], limits the possible duration between

air injection gestures, and is associated with an undesired audible injection noise, sometimes referred to as an "injection gulp". The effect of this noise is magnified because esophageal speakers (like tracheo-esophageal speakers) evidence low vocal intensity [RobbinsEtAl84], and frequently need amplification. This noise is undesirable for two reasons: (1) listeners and speakers find it objectionable and (2) in some speakers it can be mistaken for a speech segment, diminishing intelligibility. This research reports on work to detect the injection noise, with the aim of eliminating amplification during its production.

The algorithms proposed here are original solutions for a problem that has not been previously addressed, although considerable work has been undertaken to enhance other aspects of esophageal speech. (For examples, see [Qi90], [QiEtAl95], [MatsuiHara99]).

12.2 The Proposed Device

Air injection is required prior to the start of every utterance, and typically occurs again after every pause before an utterance continues. Consider a device which is used to amplify the esophageal speaker's speech. It is possible to switch amplification on after injection noise has occurred and subsided, so that the following utterance alone is transmitted. We can then switch amplification off after a period of silence has occurred, in anticipation of the next injection noise. A gain control is set to either one or zero depending on whether injection noise has been detected with an associated silence. This device (figure 12.1), which could be in a speaker's external prosthesis, or integrated with a telephone, acts to automatically remove undesired injection noise, and transmit actual speech without interruption.

12.3 Detection of Noise by HMMS

12.3.1 Feature analysis

We first proposed relatively straightforward speech recognition and word-spotting methods for detection of injection noise. [1] We treated the injection noise as a word to be spotted within an utterance spoken by an esophageal speaker. The basic scheme is shown in figure 12.2.

The signal is digitally sampled at 20 kHz. One copy of the signal is pre-emphasized and is used for processing, while a second copy is switched on or

[1] This research emerged as a collaboration between the author, and the linguist Dr. Hector Javkin, then of the University of California, Santa Barbara. Dr. Javkin introduced me to the problem and acquired the data for the study. We jointly proposed ASR methods, and I implemented them and evaluated the results. Together with the linguist Dr. Nancy Niedzielski, who hand-labeled the data in the study, we published our experimental results in the 1997 ICASSP conference proceedings.

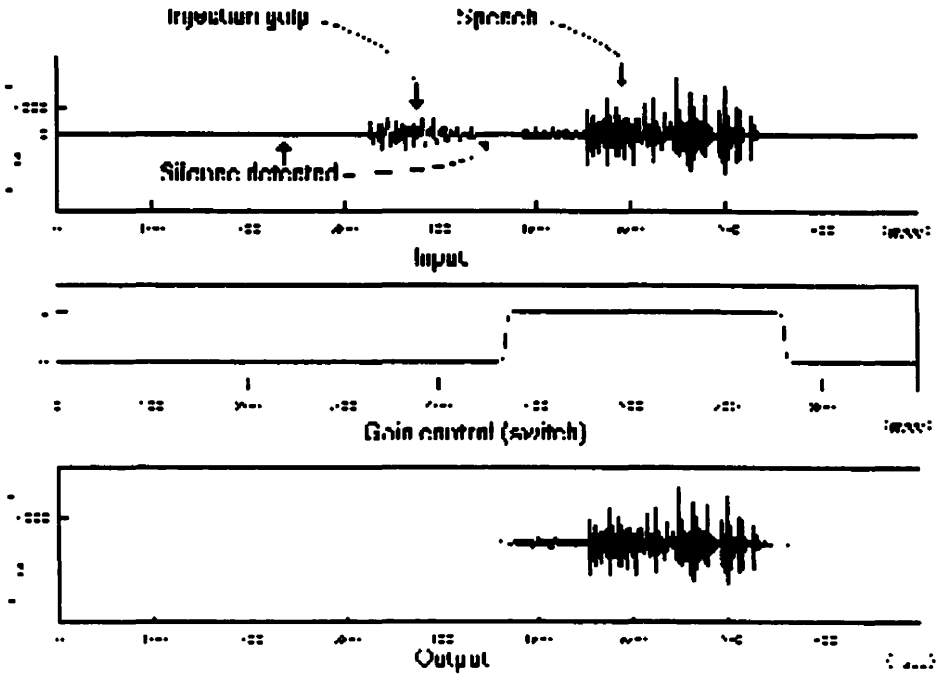

Figure 12.1: Illustration of method for rejecting injection noise.

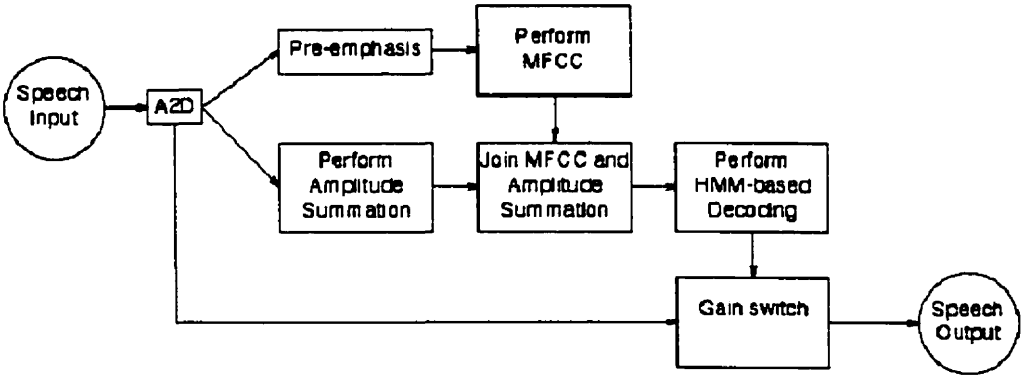

Figure 12.2: Detecting the "gulp" with HMM-based word-spotting.

off depending on the analysis. Every 10 ms. a 256-point FFT computation is performed on a 20 ms. window of speech samples. The first 12 Mel-frequency cepstral coefficients (MFCC) are calculated; these form the first part of the feature vector for a speech frame.

This spectral information is supplemented by additional information about rate of change of spectral features, consisting of the derivatives (i.e. difference cepstra). All together, 24 Mel-based cepstral coefficients are extracted from each window of the speech signal.

Time waveform analysis supplements the cepstral analysis. Specifically, a measure of signal energy is computed, along with the energy rate-of-change, based on a linear regression of 9 successive samples. The speech vector is further augmented with two extra feature points based on some special characteristics of the injection noise. When a voiced speech signal begins, it produces a negative pressure pulse. The injection noise, on the other hand, begins with a positive pressure pulse. The difference between the initial negative pressure pulse of speech and the initial positive pulse of the injection gulp is used as an additional cue for detecting the injection gulp.

A combination of a microphone, amplifiers and an analog converter is used to provide a non-inverted signal. This is done either by utilizing an even number of inverting amplifiers or by testing for an inverted signal and adding an inverting amplifier if necessary.

One of the features used to detect the polarity difference between injection noise and speech we call *amplitude summation* (AS). Amplitude summation, computed once per 10 ms. speech window, is a way to detect the initial deviation from zero in the speaker's signal. The digitized waveform is summed over intervals ranging from 1 to 20 milliseconds, depending on an adjustment for individual speakers. The probability that an injection gulp has occurred is greater when a positive value over a given threshold occurs in the summed signal. This threshold can be adjusted, depending on the associated microphones and amplifiers used to record the signal.

A second measure for detecting the polarity is obtained by differencing the center-clipped signal. To remove low-amplitude ambient noise, the signal is center clipped. The remaining signal is then differenced, to obtain the first derivative, which is then smoothed with a running average. A positive value on the result, immediately following a zero value, tends to indicate the presence of injection noise, while a negative value tends to indicate the presence of speech.

The three measures of signal energy, energy rate of change, and amplitude summation are added to the 24 Mel coefficients, to make up the complete observation vector. Thus, the acoustic front-end program creates a 27-component observation vector to represent the features of each speech frame.

12.3.2 Decoding

A hidden Markov model (HMM)-based speech decoder is used to find the optimal alignment of the speech signal with a set of speech tokens. [2] Two methods are described.

In method one, five speech tokens are used, including *silence*, *gulp*, *noise1*, *noise2* and *speech*. (The two 'noise' tokens were not fully explained, but appear to be artifacts of esophageal speech.) In method two, the speech token is replaced by a set of units representing the basic phonemes of the language. This method has more discriminative power for increased accuracy, but requires more computation.

Each token is modeled with an HMM. The number of nodes in the HMM units varies from 3, in the case of simple models such as silence, to as many as seven for certain phonemes. The number of Gaussian densities per mixture may be varied from 6 to 18 or more, depending on the limits placed on computation time by the envisioned application.

In the first implementation, five continuous mixture-density speaker-dependent HMMs were trained on a subset of a corpus of esophageal speech data, segmented and pre-labeled by hand. The HMMs contained from 3 to 7 states, with 8 Gaussian densities per mixture. The training procedure was initialized by training two models on an 8 kHz database of normal speakers: a speech model and a silence model. The distributions of these HMMs were then used to initialize the three other units. The five HMMs were then re-trained on the training half of the esophageal speech signals for a given speaker, 42 recordings in all, using Baum-Welch re-estimation. This stage of speaker-dependent training consisted of two iterations of isolated segment training and two iterations of embedded training.

The HMM decoder program decodes the speech signal frame-synchronously, with a 10 ms. frame rate. Each signal is processed by a front-end program into a vector of speech frames, as described in section 12.3.1. The Viterbi algorithm is used to estimate the conditional probabilities of the feature vectors, given each of the speech token models.

Those segments for which the injection noise (gulp) token have been labeled as most likely present, are classified as gulps within the speech signal. The remaining esophageal speech segments are transmitted, with a short processing delay, and amplified. When an injection gulp is detected, amplification is set to zero, so that it is suppressed.

---

[2] This procedure is the subject of a U.S. patent (pending) entitled "Enhancement of Esophageal Speech by Injection Noise Rejection" by Hector Javkin, Michael Galler and Nancy Niedzielski.

Number of Speech Units	239
Number of Injection Gulps	72
Gulp Detection Error-Rate	33.3%
Speech Misclassification Error	5.4%

Table 12.1: HMM results for injection noise detection.

12.4 Results of Detection of Noise by HMMS

The injection noise method was applied to a test set of utterances on which the HMMs were not trained, but from the same speaker. The results are reported in table 12.1. Two thirds of injection noise, or gulp, events were detected successfully on the speaker. Of valid speech segments, 5.4% of them were at least partially incorrectly aligned with the gulp token (speech misclassification error). These results were obtained on 40 test sentences of one speaker. Although it is likely that these results can be improved by the use of more training data and further tuning of the recognition algorithm, some of the spectral characteristics of the injection noise led to the exploration of a different approach.

12.5 Detecting Esophageal Injection Noise by Morphological Filtering

A different method for injection noise detection has been developed, based on the observation that the noise, which is produced by a gesture with a closed vocal tract, has a strong, low-frequency emphasis. This characteristic appears to be due to a double closure in the vocal tract of at least some speakers, which strongly attenuates high frequencies.

This algorithm uses a faster computation, and has a higher injection-noise detection rate on the limited data available. Through fine-tuning, it may be further improved. The data is sampled at 8 kHz. A 256-point FFT is computed every 10 ms. and smoothed by a *morphological filter* ([PitasVenetsanopoulos90], [Hansen94]) with a 10 point sliding window, removing all but the gross features of the spectral curve. Figure 12.3 shows the magnitude spectrum from the center of an injection noise segment, and the result of the morphological filter applied to the spectrum. Figure 12.4 shows the magnitude spectrum from the center of the consonant /d/ (the segment spectrally closest to an injection noise segment) and the output of the morphological filter (MF).

The mean and the derivative of the filtered spectrum are computed. The location and value of the two largest peaks are identified. A signal segment is identified as injection noise if the following criteria are met:

1. The largest peak is lower in frequency than the second largest peak.

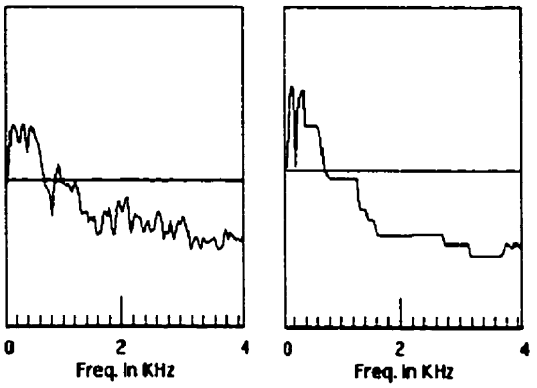

Figure 12.3: 256-point FFT from the center of an injection noise segment, and the result of passing the FFT through the morphological filter.

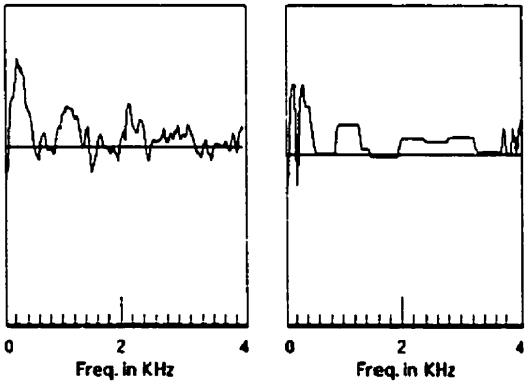

Figure 12.4: 256-point FFT from the center of a /d/ segment and the result of passing the FFT through the morphological filter.

Number of Speech Units	235
Number of Injection Gulps	79
Gulp Detection Error-Rate	17.7%
Speech Misclassification Error	15.7%

Table 12.2: Morphological filter experiment: results on data set 1.

Number of Speech Units	242
Number of Injection Gulps	72
Gulp Detection Error-Rate	16.7%
Speech Misclassification Error	17.4%

Table 12.3: Morphological filter experiment: results on data set 2.

2. All points above 2 kHz are less than the mean.

The second method [3] was tested on both the training and test data sets used for the HMMs in the first method. Again, results were encouraging, although the speech misclassification error was unacceptably high.

12.6 Discussion

The results obtained this far were achieved with relatively untuned algorithms. Furthermore, no attempt has been made at present to combine some of the features used in the HMM-based method with those used with the method based on morphological filtering. Such a combination would likely reduce the error. On the basis of these results and the likelihood that they can be improved, injection gulp rejection could work in a way totally transparent to the user, by means of an electronic switch that would only turn amplification on after a gulp and a following short silence have occurred. Whenever a speaker paused, the amplification would be turned off, waiting for the injection gulp before turning it on again. It could, in theory, work without any delay in the output signal.

Adjustments have to be made for speakers who use multiple gulps in order to sufficiently insufflate the esophagus. If a speaker consistently used double or triple gulps, the method could be tuned to reject them. However, speech with varying numbers of gulps could be problematic.

The methods and application described here deal with only one aspect of esophageal speech. Research has continued in ways to improve other aspects as well. For example, Japanese researchers have attempted to improve the overall quality of the speech by resynthesizing the voicing source of esophageal speech using formant-based analysis and re-synthesis. Subjective

[3]The subject of another U.S. patent filing, docket no. MATI-A244 filed April 16, 1997, entitled "Esophageal Speech Injection Noise Detection and Rejection" by Hector Javkin, Michael Galler, Nancy Niedzielski, and Robert Boman.

ratings by speech therapists scored the the synthesized speech higher than the original [MatsuiHara99].

Part IV

Conclusion

Chapter 13

Perspectives

13.1 Trends in Speech Recognition Research

To know the state of speech science in 1999, and the steady ferment of activity within the speech research community, we need look at the recent published work of a relatively small number of research groups that have long made contributions to the state of the art. These groups include AT&T labs; Germany's Aachen University of Technology; the Cambridge, Massachusetts firm BBN Technologies; P.C. Woodland's and Tony Robinson's research groups at Cambridge University; Carnegie Mellon University's Interactive System Laboratories; the Newton, Massachusetts firm of Dragon Systems; IBM's Thomas Watson Research Center; the French LIMSI-CNRS Spoken Language Processing group; MIT's Spoken Language Systems Group; and the firm SRI International of Menlo Park, California. This list is not exhaustive. The Oregon Graduate Institute, ICSI at Berkeley, and several government funded research institutes including ATR in Japan, IRST in Italy, and CRIM in Quebec, also belong in the list of groups making important contributions to the field over a long period of time.

In addition, the electronic giants Philips, Motorola, Matsushita, and Sony, the Belgian firm Lernout and Hauspie, Finland's Nokia, the U.S. firm Nuance Communications and the Microsoft Research Group are notable for their speech recognition and synthesis technology, although for business reasons most of these companies are not prolific publishers of original research work.

By examining recent work of these groups it is possible to generalize about the techniques and areas of study which are being most actively pursued in 1999. What follows is a broad, if not exhaustive, survey of current themes, and subjects of interest in speech recognition research.

Managing robustness issues. Researchers have gained a great deal of hard experience in dealing with the mismatch between training conditions and test conditions. Every practitioner has had the experience of transferring a successful recognizer from its development environment to try it in a new

room, with new speakers, and perhaps a different microphone, with results often much less than we hope. This is the "robustness" issue.

We now have a toolkit of reliable techniques for channel normalization and adaptation. Systems that employ cepstral feature analysis invariably apply the technique of Cepstrum Mean Normalization (CMN). An average MFCC vector (section 5.4.6) is computed over a whole input sentence or segment, and subtracted from each of the frames in the segment. This serves to reduce the effects of constant channel characteristics. The same normalization can be applied in perceptual linear predictive (PLP) analysis [Hermansky90]. PLP analysis convolves a set of critical band filters with the speech spectrum. These modify the spectrum according to perceptual measurements of the auditory system and lead to PLP-Cepstra feature sets. In [WoodlandEtAl97] and [BakisEtAl97] there is evidence that PLP alone is more robust to environmental mismatch than MFCC. Cepstral filtered PLP or Mel-filtered PLP is more robust than either. CMN can be applied to both these feature sets. Cepstral *variance* normalization has also been tried with some success in conversational speech [HainEtAl99].

Cepstral Mean Normalization can reduce the word error rate on **ATIS** by an absolute 1.1% [BocchieriEtAl95]. If a constant bias is observed in the training or test recordings, another small improvement can be achieved by removing this bias before speech analysis.

Channel variation in telephone speech is effectively managed with the use of RASTA filtering. The MFCC feature vector is not robust in the presence of channel distortions and background noise. RASTA filtering is an additional front end operation which, combined with MFCC or PLP analysis, actively reduces channel effects and distortion due to background noise [HermanskyEtAl95]. In the log power spectral domain channel distortions are additive. However most channels are stationary, or very slowly varying, and impart a near constant offset on the time series of short-time log power spectral vectors. Applying a sharp cut-off highpass filter to each of the spectral bins, across time, removes this slowly varying offset and supresses the channel distortion.

The RASTA filter is also able to partially mask the effects of background noise. The assumption is that the speech portion of a signal is relatively stationary compared to the noise. By lowpass filtering the time series of log spectral vectors the effects of noise are reduced. To combine both the highpass and lowpass advantages of filtering the vector stream RASTA is implemented as an IIR (infinite-duration impulse response) bandpass filter. The designers of a 1264 isolated word recognition task, a telephone autoattendant, achieved a 7.2% improvement in recognition by adding the RASTA filter to the front end analysis [AzzopardiEtAl98].

All the techniques above attack the robustness problem by reducing the effects of noise during the front-end signal analysis. In addition to normalization methods and noise-robust feature extraction, there is also a vigorous literature exploring how to compensate within the recognition unit for

mismatched conditions. The same techniques that have been developed for speaker adaptation are applied to environmental adaptation. Good results have also been achieved with Gales' method called Parallel Model Combination (PMC) [GalesYoung93], in which additive and convolutional noise are explicitly modeled and used to transform the initially clean HMM model space.

The innovation of sub-band based speech recognition. Studies of auditory perception have long suggested that human beings decode information from different frequency subbands independently [Allen94]. This suggests that a correct linguistic interpretation is possible if some subbands contain correct information even if other subbands are corrupted by noise. To put it differently, the human auditory mechanism may be capable of de-emphasizing unreliable sub-bands, or performing "missing feature" compensation. In the last three years a number of researchers have been exploring sub-band based recognition in hybrid HMM/ANN (artificial neural network) systems. In this work, the full auditory frequency band has been divided up into an arbitrary number of sub-bands, e.g. 2, 4 or 7. Each of these bands are independently subjected to feature analysis, and phoneme probabilities are computed for each. The results are then recombined at some segmental level, using dynamic programming or a neural net, in order to produce a single hypothesis. The evidence is that sub-band recognition is as least as effective as full-band recognition on clean speech. In a test on clean speech from the **Switchboard** corpus, the multi-band recognizer reduced the error rate from 63.6% to 61.4% [BourlardDupont97]. Admittedly, the baseline system performs badly enough that it is hard to draw conclusions. But other studies also found that the sub-band recognizer is at least as good as its full-band counterpart on clean speech - see [TibrewalaHermansky97]. Additionally, the multi-band recognizer degrades more gracefully when noise is localized in parts of the spectrum. However, when noise is present across all the sub-bands, the sub-band decoder may underperform a baseline system [TibrewalaHermansky97],[OkawaEtAl98].

Intriguingly, multi-band recognition also offers the potential of combining information at multiple time resolutions.

The introduction of vocal tract length normalization (VTLN). Differences in vocal tract length of speakers results in a warping of the frequency axis in observed formant energy displacements. These differences contribute to speaker feature variability, and are obviously correlated with gender, and age (children vs. adults). A common, if brute force approach to manage this variability is to train gender-dependent models, and when training data permits, age-dependent models. VTLN tries to compensate directly for these differences by warping the frequency axis such that formant locations remain stationary across speakers. VTLN has been shown to achieve a 1.4% absolute reduction in word error rate on a gender balanced subset of the **Switchboard**

corpus [BillaEtAl99]. Dragon Systems did a study to contrast the benefits of VTLN with conventional gender-training on the **Broadcast News** corpus. They found an identical 2.2% absolute reduction in the baseline 39.0% word error rate [WegmannEtAl99]. Combining the two techniques provides a further small improvement, but this improvement vanishes when unsupervised rapid adaptation is applied to models trained with either technique alone.

The importance of including fast unsupervised speaker adaptation in the final recognition pass. The evolution of speech recognition has traced an arc from initial systems, which were all speaker-dependent, to the speaker-independent systems that predominated until recently. At present the circle has closed; a strong component of speaker adaptation is added to every prominent research system. The motivation for this new emphasis springs from two realizations. First, the training of speaker-independent models mixes many kinds of variability in an undifferentiated way, as if they represented the same dimension in feature space. For example, two speakers may characteristically pronounce a phoneme differently within a given phonetic context, yet the mixtures of a single triphone are required to represent both variations. Second, adding more samples from new training data will not solve this problem. Rather than better separate the cluster from its neighbors in model-state space, more training may only further smear the cluster through the space.

We have discussed techniques for channel normalization, for both constant channel characteristics, and slowly varying ones. Also normalization techniques for background noise, for the speaker's sex, and vocal tract length. If follows that however brief the segment to be recognized, a good speaker-independent system will have the capacity to normalize for the current speaker's vocal characteristics. What is needed is a method of rapid unsupervised speaker adaptation before the final output hypothesis is formulated.

The list of important large vocabulary projects incorporating rapid unsupervised speaker adaptation in some recognition pass is inclusive: BBN's **Byblos**, Dragon's **Broadcast News** transcriber, Cambridge University's **HTK**, Carnegie Mellon's **Janus** speech engine, IBM's **LVCSR**, and LIMSI's large vocabulary recognizer all make use of it.

The BBN system is a good case study [BillaEtAl99]. Recognition on data from the **Switchboard** and English **Callhome** corpora is carried out in five stages:

1. A speaker's gender and VTL parameter are estimated by standard modeling techniques.

2. Transcriptions are generated with speaker independent models.

3. Maximum Likelihood Linear Regression (MLLR) is used to adapt the means and variances based on these imperfect transcriptions.

4. New N-best transcriptions are generated with the speaker adapted models.

5. The results of stage 4 are rescored with more detailed language models to find the single best transcription.

A typical improvement due to unsupervised adaptation, as measured on the **Broadcast News** corpus, reduces the word error rate from 39.9% to 37.7% [WegmannEtAl99]. Similar results are achieved on telephone speech [PeskinEtAl99].

Fast likelihood computation. A small speech decoder for embedded applications, based on design principles described in Chapter 10, distills the necessary work down to little more than a repetitive series of emission probability calculations. The speed of the decoder will thus depend on the Gaussian computations. At the other end of the scale, a large research system will target vocabularies on the order of 20,000 to 60,000 words, and run up to 500 times real time to achieve maximum accuracy. In spite of earlier successes with such tasks as read speech and dictation, researchers continue to grapple with a heavy computational burden as the scope and ambition of the recognition problem expands to include real conversational speech, typified by the *Switchboard* and *CallHome* corpora. These research systems may employ on the order of 140,000 Gaussian or Laplacian densities [NeyEtAl98] to achieve accurate recognition. BBN reports that in their BYBLOS decoder, when a wide search beam is employed for maximum accuracy 93% of the work in the second pass is Gaussian computation. When a tight beam is employed for efficiency, 76% of the first pass and 94% of the second pass is busy with calculating likelihoods [DavenportEtAl99].

Early efforts at speeding up this step were based on vector quantization (VQ) of the speech feature space. The Gaussian means were grouped into clusters, using *K-means* or similar algorithms. The centroids of the clusters formed a codebook, and for each frame of speech, a distance computation was performed between the input feature vector and each of the codewords. This would identify the cluster to which the vector belonged, and likelihood computations were only be performed for distributions belonging to the cluster.

My own early experiments proved the accuracy/speedup tradeoff does not favour this simple VQ approach. Similarly, others have shown that applying Linear Discriminant Analysis (LDA) to find the most significant components of the feature vector for selecting the most relevant distributions had a poor error/speed tradeoff. However, the VQ method can provide a "coarse preselection of the prototype vectors of the densities". In [NeyEtAl98] the systems computes only those prototype vectors located inside a hypercube centered at the input feature vector, and achieves 75% reduction in recognition time for a 20,000 word vocabulary with little loss in accuracy.

In 1997 researchers at IBM T.J.Watson Center proposed a decision-tree procedure to quantize the feature space. The space is hierarchically divided into disjoint regions separated by the intersection of hyperplanes. In each of these regions resides a subset of the allophone set. In training the decision tree, the data at each node is repartitioned into two child nodes such that the average entropy of the allophone distributions at the new nodes is minimized. Traversal of the decision tree is efficient, requiring an inner product calculation of the hyperplane with the feature vector at each node. Execution time is reduced by 95% with insignificant loss of accuracy [PadmanabhanEtAl97]. A simpler and nearly as robust variation of the decision-tree method repartitions the nodes with binary clustering [DavenportEtAl99].

13.2 Open Problems and Future Directions

We who toil in the field of computer speech recognition are full of reasons why the full problem is so very difficult to solve. Speech and reason are the two faculties which make us human – they are *that* fundamental. It is not even clear that they can exist separately; that a thought can exist apart from the language, however personal, ephemeral and silent, that the mind uses to conjure it. How can anything be more difficult for us than the processing of language?

On a prosaic level, we list all the sources of variability that preclude a simple algorithm for correctly mapping recorded speech to text. This familiar list includes the imprecision of human vocal production, the variability due to differences in vocal tract, in dialect, in emotional context, in speaking style, in characteristics of transmission channel. There are the natural disfluencies of conversational speech, with its hesitations, restarts, and abundance of non-speech utterances. Then there are the effects of noise, both speech and non-speech, which may overlap or obscure the signal of interest. People can focus in on one particular speech signal amid many.

There are huge and open-ended vocabularies to deal with, inconsistencies in pronunciation, both valid and invalid ones, and an infinity of plausible utterances. Finally, when a fragment of speech is correctly decoded into a sequence of phonemes, there may not be a unique mapping to text. It may require syntactic, semantic and contextual linguistic information to make the translation to the right sequence of words.

Robustness in communication is achieved by a large expansion in bandwidth. Speech is an example: the underlying sound patterns have a bandwith of perhaps 30 Hz, but vocal tract encodes them in an 8 kHz bandwith [Atal99]. This large redundancy compensates for the imprecision of speech production and the distortions introduced by the communication channel. Clearly the human auditory/mental apparatus is marvelously robust in making sense of a speech signal. Humans can accurately recognize speech that is

degraded by environmental acoustics and noise, speech reproduced through bandwidth- limited and noisy transmission channels, channels with erratic linear frequency response, as well as speech that is high-pass and/or low-pass filtered [LippmannCarlson97].

So we might turn the question around like this: why is human speech recognition so easy? How do we process so effectively the systematic variabilities encoded in that highly redundant signal?

Martin Russell argued in a 1997 IEEE Workshop that data-driven modeling of speech with HMMs attempts to characterize the surface structure of speech and only superficially models its underlying mechanisms.

> Variability, which might be explicable with reference to this deeper level, manifests itself as as intricate and complex change in the surface structure. Faced with this surface complexity and no framework for modeling the underlying causality, the simplest solution, which is adopted in HMMs, is to assume that this surface variation is random.

I believe this point of view is *not* controversial in the speech fraternity. In spite of all the successes, computer speech recognition is in a primitive state and we know this. The systems that work today may perform impressive tasks, but that competence has been bought with simple mechanisms for learning to classify sounds base on massive (and painstakingly assembled) collections of labeled data.

Future breakthroughs will involve the discovery of intermediate levels at which to model descriptions of the speech process. They will integrate multi-resolutional trajectories of speech parameters, some of which are yet unknown. A larger proportion of parameters in the speech model will be explicitly estimated with the help of new underlying models of speech production and perception; fewer parameters will be empirically estimated.

The future speech engine may still incorporate large amounts of linguistic information, but the core decoder will be engineered more simply and compactly. And it may at last achieve the same generality as exhibited by a hearing child. It will be able to recognize all sorts of speech, from heterogeneous sources, filtered in a variety of ways, without the need for extra training or adaptation.

Bibliography

[AbrashEtAl96] Abrash V., Sankar A., Franco H., and Cohen M., "Acoustic Adaptation Using Nonlinear Transformations of HMM Parameters", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 729–732.

[AceroHuang96] Acero A., and Huang X., "Speaker and Gender Normalization for Continuous-Density Hidden Markov Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 342–345.

[AckleyEtAl85] Ackley D.H., Hinton G.E., and Sejnowski T.J., "A Learning Algorithm for Boltzmann Machines", *Cognitive Science*, 1985, 9, pp. 147–169.

[Adda-DeckerEtAl99] Adda-Decker M., Adda G., Gauvain J.-L., and Lamel L., "Large Vocabulary Recognition in French", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 45–48.

[AliEtAl98] Abdelatty Ali A.M., der Spiegel J.V., and Mueller P., "An Acoustic-phonetic Feature-based System for the Automatic Recognition of Fricative Consonants", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 961–964.

[Alkhairy99] Alkhairy A., "An Algorithm for Glottal Volume Velocity Estimation", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 233–236.

[Allen94] Allen J.B., "How do humans process and recognize speech?", *IEEE Transactions on Speech and Audio Processing*, 1994, Vol. 2, pp. 567–577.

[Alleva97] Alleva F., "Search Organization in the Whisper Continuous Speech Recognition System", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 295–302.

[AllevaEtAl96] Alleva F., Huang X., and Hwang M-Y., "Improvements on the Pronunciation Prefix Tree Search Organization", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 133–136.

[AnastasakosEtAl97] Anastasakos T., McDonough J., and Makhoul J., "Speaker Adaptive Training: A Maximum Likelihood Approach to Speaker Normalization", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1043–1046.

[AntoniolEtAl94] Antoniol G., Brugnara F., Cettolo M., and Federico M., "Language Model Estimations and Representations for Real-Time Continuous Speech Recognition", *Proceedings of the International Conference on Speech and Language Processing*, September 1994.

[AsadiEtAl95] Asadi A., "Combining Speech Algorithms into a "Natural" Application of Speech Technology for Telephone Network Services", *Eurospeech 95*, pp. 273-276.

[Atal99] Atal B.S., "Automatic Speech Recognition: A Communication Perspective", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 457–460.

[AzzopardiEtAl98] Azzopardi D., Milner B., and Semnani S., "Improving the Accuracy of Telephony based Speaker Independent Speech Recognition", *The International Conference on Signal Processing Applications and Technology*, 1998.

[BacchianiEtAl96] Bacchiani M., Ostendorf M., Sagisaka Y., and Paliwal K., "Design of a Speech Recognition System Based on Acoustically Derived Segmental Units", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 443–446.

[Bahl86] Bahl L.R., Brown P.F., de Souza P.V., Nahamoo D., and Mercer R.L., "Maximum Mutual Information Estimation of Hidden Markov Models Parameters for Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1986, pp. 49–52.

[BahlEtAl91] Bahl L.R., deSouza P.V., Gopalakrishnan P.S., Nahamoo D., and Picheny M.A, "Decision Trees for Phonological Rules in Continuous Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1991, pp. 185-188.

[BahlEtAl96] Bahl L.R., Padmanabhan M., Nahamoo D., and Gopalakrishnan P.S., "Discriminative Training of Gaussian Mixture Models for Large Vocabulary Speech Recognition Systems", *Proceedings of the*

IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996, pp. 613–616.

[BakamidisEtAl90] Bakamidis S., Dendrinos M., and Carayannis G., "SVD Analysis by synthesis of harmonic signals", *IEEE Transactions on Signal Processing*, 1990, Vol. 39, No. 2, pp. 472–476.

[Baker75] Baker J.K., "Statistical Modeling as a Means of Automatic Speech Recognition", Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, 1975.

[BakisEtAl97] Bakis R., Chen S., Gopalakrishnan P., Gopinath R., Maes S., and Polymenakos L., "Transcription of Broadcast News - System Robustness Issues and Adaptation Techniques", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 711–714.

[BasuEtAl99] Basu S., Micchelli C.A., and Olsen P.A., "Maximum Likelihood Estimates for Exponential Type Density Distributions", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 361–364.

[Baum72] Baum L.E. "An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes", *Inequalities*, 1972, Vol. 3, pp. 1–8.

[BeaufaysEtAl99] Beaufays F., Weintraub M., and Konig Y., "Discriminative Mixture Weight Estimation for Large Gaussian Mixture Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 337–340.

[BeigiEtAl98] Beigi H.S.M., Maes S.H., and Sorenson J.S., "A Distance Measure between Collections of Distributions and its Application to Speaker Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 753–756.

[Bellegarda98] Bellegarda J.R., "Exploiting Both Local and Global Constraints for Multi-span Statistical Language Modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 677–680.

[Bellegarda99] Bellegarda J.R., "Speech Recognition Experiments Using Multi-Span Statistical Language Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 717–720.

[BengioEtAl90] Bengio Y., Cardin R., De Mori R., and Normandin Y., "A Hybrid Coder for Hidden Markov Models Using a Recurrent Neural

Network", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1990, pp. 537-540.

[BergerMiller98] Berger A., and Miller R., "Just-in-Time Language Modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 705–708.

[BeulenEtAl99] Beulen K., Ortmanns S., and Elting C., "Dynamic Programming Search Techniques for Across-Word Modeling in Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 609–612.

[BeulenNey98] Beulen K., and Ney H., "Automatic Question Generation for Decision Tree Based State Tying", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 805–808.

[Beyelein97] Beyerlein P., "Discriminative Model Combination", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 238–245.

[Beyerlein98] Beyerlein P., "Discriminative Model Combination", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 481–484.

[Billa97] Billa J., "Dual-channel Auditory Spectrum Modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 995–998.

[BillaEtAl99] Billa J., Colhurst T., El-Jaroudi A., Iyer R., Ma K., Matsoukas S., Quillen C., Richardson F., Siu M., Zavaliagkos G., and Gish H., "Recent Experiments in Large Vocabulary Conversational Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 41–44.

[BilliEtAl95] Billi R., "Interactive Voice Technology at Work: The CSELT Experience", *Speech Communication*, 17, pp. 263–271, November 1995.

[Blasig99] Blasig R., "Combination of Words and Word Categories in Varigram Histories", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 529–532.

[Block62] Block H.D., "The Perceptron: a Model for Brain Functioning", *Reviews of Modern Physics*, 1962, 34, pp. 123–135.

[Bocchieri93] Bocchieri E., "A Study of the Beam-Search Algorithm for Large Vocabulary Continuous Speech Recognition and Methods for Improved Efficiency", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1993, pp. 1521–1524.

[BocchieriEtAl95] Bocchieri E., Riccardi G., and Anantharaman J., "The 1994 AT&T CHRONUS Recognizer", *Proceedings of the ARPA Human Language Technology Workshop*, 1995, pp. 265–268.

[BocchieriEtAl99] Bocchieri E., Digalakis V., Corduneanu A., and Boulis C., "Correlation Modeling of MLLR Transform Biases for Rapid HMM Adaptation to New Speakers", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 773–776.

[BourlardDupont97] Bourlard H., and Dupont S., "Subband Based Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1251–1254.

[BourlardEtAl94] Bourlard H., Dhoore B., and Boite J.-M., "Optimizing Recognition and Rejection Performance in Wordspotting Systems", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1994, pp. 373–376.

[Brill98] Brill E., "Machine Learning and Automatic Linguistic Analysis: The Next Step", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 1033–1036.

[Buntine] Buntine W.L., "Theory Refinement of Baysian Networks", *Seventh Conference on Uncertainty in Artificial Intelligence*, Anaheim CA.

[ByrneEtAl97] Byrne B., Finke M., Khudanpur S., McDonough J., Nock H., Riley M., Saraclar M., Wooters C., and Zavaliagkos G., "Pronunciation Modeling for Conversational Speech Recognition: A Status Report from WS97", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 26–33.

[ByrneEtAl98] Byrne W., Finke M., Khudanpur S., McDonough J., Nock H., Riley M., Saraclar M., Wooters C., and Zavaliagkos G., "Pronunciation Modeling Using a Hand-labelled Corpus for Conversational Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 313–316.

[Casacuberta90EtAl] Casacuberta F.,Vidal E., Mas B., and Rulot H., "Learning the Structure of HMMs through Grammatical Inference Techniques", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1990, pp. 717–720.

[ChangLippmann96] Chang E.I., and Lippmann R.P., 'Improving Wordspotting Performance with Artificially Generated Data", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 526–529.

[ChenEtAl98] Chen S.F., Seymore K., and Rosenfeld R., "Topic Adaptation for Language Modeling Using Unnormalized Exponential Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 681–684.

[ChenEtAl99] Chen S.S., Eide E.M., Gales M.J.F., Gopinath R.A., Kanevsky D., and Olsen P., "Recent Improvements to IBM's Speech Recognition System for Automatic Transcription of Broadcast News", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 37–40.

[ChenRosenfeld99] Chen S.F., and Rosenfeld R., "Efficient Sampling and Feature Selection in Whole Sentence Maximum Entropy Language Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 549–552.

[Chengalvarayan99] Chengalvarayan R., "Hierarchical Subband Linear Predictive Cepstral (HSLPC) Features for HMM-based Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 409–412.

[ChongEtAl98] Chong N.R., Burnett I.S., Chicaro J.F., and Thomson M.M., "Use of the Pitch Synchronous Wavelet Transform as a New Decomposition Method for WI", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 513–516.

[Chou97] Chou W., "Minimum Error Rate Training for Designing Tree-Structured Probability Density Function", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1507–1510.

[ChouReichl99] Chou W., and Reichl W., "Decision Tree State Tying Based on Penalized Bayesian Information Criterion", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 345–348.

[Chow90] Chow Y.L., "Maximum Mutual Information Estimation of HMM Parameters for Continuous Speech Recognition using the N-Best Algorithm", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1990, pp. 701–704.

[ChowEtAl86] Chow Y.L., Schwartz R., Roucos S., Kimball O., Price P., Kubala F., Dunham M., Krasner M., and Makhoul J., "The Role of Word-Dependent Coarticulatory Effects in a Phoneme-Based Speech Recognition System", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1986, pp. 1593–1596.

[ClarksonMoreno99] Clarkson P., and Moreno P.J., "On the Use of Support Vector Machines for Phonetic Classification", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 585–588.

[ClarksonRobinson97] Clarkson P.R., and Robinson A.J., "Language Model Adaptation Using Mixtures and an Exponentially Decaying Cache", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 799–802.

[CohenEtAl95] Cohen M., Rivlin Z., and Bratt H., "Speech Recognition in the ATIS Domain Using Multiple Knowledge Sources", *Proceedings of the ARPA Human Language Technology Workshop*, 1995, pp. 257–260.

[CohenEtAl97] Cohen P., Dharanipragada S., Gros J., Monkowski M., Neti C., Roukos S., and Ward T., "Towards a Universal Speech Recognizer for Multiple Languages", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 591–598.

[ColeEtAl91] Cole R., Roginski K., and Fanty M., "English Alphabet Recognition with Telephone Speech", *Eurospeech 91*, pp. 479–482.

[ColeEtAl96] Cole R.A., Yan Y., Mak B., Fanty M., and Bailey T., "The Contribution of Consonants Versus Vowels to Word Recognition in Fluent Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 853–856.

[CookEtAl96] Cook G.D., Christie J.D., Clarkson P.R., Hochberg M.M., Logan B.T., and Robinson A.J., "Real-Time Recognition of Broadcast Radio Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 141–144.

[CookRobinson98] Cook G., and Robinson T., "Transcribing Broadcast News with the 1997 ABBOT System", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 917–920.

[CookeEtAl93] Cooke M., Beet S., and Crawford M., editors, *Visual Representations of Speech Signals*, John Wiley & Sons, New York NY, 1993.

[CookeEtAl97] Cooke M., Morris A., and Green P., "Missing Data Techniques for Robust Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 863–866.

[CorazzaEtAl91] Corazza A., De Mori R., Gretter R., and Satta G., "Optimal Probabilistic Evaluation Functions for Search Controlled by Stochastic Context-free Grammars", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[CorrazzaEtAl91] Corazza A., De Mori R., Gretter R., anbd Satta G., "Optimal Probabilistic Evaluation Functions for Search Controlled by Stochastic Context-free Grammars", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[Corredor-ArdoyEtAl98] Corredor-Ardoy C., Lamel L., Adda-Decker M., and Gauvain J.L., "Multilingual Phone Recognition of Spontaneous Telephone Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 413–416.

[CoxRose96] Cox S., and Rose R.C., "Confidence Measures for the Switchboard Database", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 511–514.

[DahlEtAl95] Dahl D.A., Norton L.M., Weir C.E., and Linebarger M.C., "Weakly Supervised Training for Spoken Language Understanding Systems", *Proceedings of the ARPA Human Language Technology Workshop*, 1995, pp. 272–275.

[DavenportEtAl99] Davenport J., Schwartz R., and Nguyen L., "Towards a Robust Real-Time Decoder", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 645–648.

[DavisMermelstein80] Davis S., and Mermelstein P., "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1980, pp. 357–366.

[Demori79] De Mori R., "Recent Advances in Automatic Speech Recognition", *Signal Processing*, 1979, **1**(2): pp. 95–124.

[Demori98] De Mori R., editor, *Spoken Dialogues with Computers*, Academic Press, San Diego CA, 1998.

[DemoriEtAl76] De Mori R., Laface P., and Piccolo E., "Automatic Detection and Description of Syllabic Features in Continuous Speech", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1976, Vol. 24, pp. 365–379.

[DemoriEtAl90] Demori R., Palakal M.J., and Cosi P., "Perceptual Models for Automatic Speech Recognition Systems", *Advances in Computers*, Vol. 31, pp. 99–173.

[DemoriEtAl95] De Mori R., Snow C., and Galler M., "On the Use of Stochastic Inference Networks for Representing Multiple Word Pronunciations", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995, pp. 701–704.

[DemoriGaller95] De Mori R., Galler M., and Brugnara F., "Search and Learning Strategies for Improving Hidden Markov Models" *Computer Speech and Language*, 9, pp. 107–121, 1995.

[DemoriGaller96] De Mori R., and Galler M., "The Use of Syllable Phonotactics for Word Hypothesization", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 877–880.

[DemoriSnowGaller95] De Mori R., Snow C., and Galler M., "On the Use of Stochastic Inference Networks for Representing Multiple Word Pronunciations", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995.

[DentonTaylor92] Denton J.S., and Taylor C.R., editors, "Final Report on Speech Recognition Research: December 1984 to April 1990", Technical Report for DARPA, School of Computer Science, Carnegie Mellon University, July 1992.

[Deng97] Deng L., "Integrated Multilingual Speech Recognition Using Universal Phonological Features in a Functional Speech Production Model", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1007–1010.

[Deng97a] Deng L., "A Dynamic, Feature-based Approach to Speech Modeling and Recognition", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 107–114.

[DeshmukhEtAl96] Deshmukh N., Weber M., and Picone J., "Automated Generation of N-Best Pronunciations of Proper Nouns", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 283–286.

[DeshmukhEtAl97] Deshmukh N., Ngan J., Hamaker J., and Picone J., "An Advanced System to Generate Pronunciations of Proper Nouns", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1467–1470.

[DharanRoukos98] Dharanipragada S., and Roukos S., "A Fast Vocabulary Independent Algorithm for Spotting Words in Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 233–236.

[DharaniRoukos97] Dharanipragada S., and Roukos S., "New Word Detection in Audio-Indexing", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 551–557.

[DigalakisEtAl98] Digalakis V., Neumeyer L., and Perakakis M., "Quantization of Cepstral Parameters for Speech Recognition over the World Wide Web", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 989–992.

[DigalakisEtAl99] Digalakis V., Berkowitz S., Bocchieri E., Boulis C., Byrne W., Collier H., Corduneanu A., Kannan A., Khudanpur S., and Sankar A., "Rapid Speech Recognizer Adaptation to New Speakers", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 765–768.

[DudaHart73] Duda R.O and Hart P.E., *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

[DupontEtAl97] Dupont S., Bourlard H., Deroo O., Fontaine V., and Boite J-M., "Hybrid HMM/ANN Systems for Training Independent Tasks: Experiments on *Phonebook* and Related Improvements", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1767–1770.

[EideGish96] Eide E., and Gish H., "A Parametric Approach to Vocal Tract Length Normalization", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 346–349.

[ElMelianiOShaugh97] El Meliani R., and O'Shaughnessy D., "Accurate Keyword Spotting Using Strictly Lexical Filters", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 907–910.

[EphDem89] Ephraim Y., Dembo A., and Rabiner L.R. "A Minimum Discrimination Information Approach for Hidden Markov Modeling", Markov Modeling", *IEEE Transactions on Information Theory*, 35, No. 5, September 1989 pp. 1001–1013.

[Ephraim96] Ephraim Y., "Gain-Adapted Hidden Markov Models for Recognition of Clean and Noisy Speech", *IEEE Transactions on Signal Processing*, Vol. 40, No. 6, 1992, pp. 1303–1316.

[Ferguson80] Ferguson J., Ed., *Hidden Markov Models for Speech*, IDA, Princeton NJ, 1980.

[FetterEtAl96] Fetter P., Kaltenmeier A., Kuhn T., and Regel-Brietzmann P., 'Improved Modeling of OOV Words in Spontaneous Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 534–537.

[FinkeRogina97] Finke M., and Rogina I., "Wide Context Acoustic Modeling in Read Vs. Spontaneous Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1743–1746.

[FischerStahl99] Fischer A., and Stahl V., "Database and Online Adaptation for Improved Speech Recognition in Car Environments", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 445–448.

[Fletcher69] Fletcher R., *Optimization*, Berkeley Square House, London, 1969.

[Forney73] Forney G.D., "The Viterbi Algorithm", *IEEE Proceedings*, 1973, 61, pp. 268–278.

[FraserEtAl96] Fraser N.M., Salmon B., and Thomas T., "Call Routing by Name Recognition: Field Trial Results for the Operetta System", *1996 IEEE Third Workshop - Interactive Voice Technology for Telecommunications Applications*, pp. 101–104.

[FritschEtAl97] Fritsch J., Finke M., and Waibel A., "Context-Dependent Hybrid HME/HMM Speech Recognition Using Polyphone Clustering Decision Trees", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1759–1762.

[FritschRogina96] Fritsch J., and Rogina I., "The Bucket Box Intersection (BBI) Algorithm for Fast Approximative Evaluation of Diagonal Mixture Gaussians", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 837–840.

[FukadaSagisaka98] Fukada T., and Sagisaka Y., "Speaker Normalized Acoustic Modeling Based on 3-D Viterbi Decoding", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 437–440.

[Gales98] Gales M.J.F., "Semi-tied Covariance Matrices", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 657–660.

[GalesYoung93] Gales M.J.F., and Young S., "Cepstral Parameter Compensation for HMM Recognition", *Speech Communication*, 1993, Vol. 12, No. 3, pp. 231–239.

[GallerJunqua97] Galler M., and Junqua J-C., "Robustness Improvements in Continuously Spelled Names over the Telephone", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1539–1542.

[GanapathEtAl97] Ganapathiraju A., Goel V., Picone J., Corrada A., Doddington G., Kirchhoff K., Ordowski M., and Wheatley B., "Syllable - A Promising Recognition Unit for LVCSR", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 207–214.

[GaoEtAl99] Gao Y., Jan E.-E., Padmanabhan M., and Picheny M., "HMM Training Based on Quality Measurement", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 129–132.

[GarciaLee97] Garcia-Mateo C., and Lee C-H., "A Study on Subword Modeling for Utterance Verification in Mexican Spanish", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 614–621.

[GauvainEtAl96] Gauvain J.L., Lamel L., Adda G., and Matrouf D., "Developments in Continuous Speech Dictation Using the 1995 ARPA NAB News Task", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 73–76.

[GauvainEtAl97] Gauvain J.L., Adda G., Lamel L., and Adda-Decker M., "Transcribing Broadcast News Shows", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 715–718.

[GeutnerEtAl98] Geutner P., Finke M., and Scheytt P., "Adaptive Vocabularies for Transcribing Multilingual Broadcast News", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 925–928.

[GeutnerEtAl99] Geutner P., Finke M., and Waibel A., "Selection Criteria for Hypothesis Driven Lexical Adaptation", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 617–620.

[GillickEtAl97] Gillick L., Ito Y., and Young J., "A Probabilistic Approach to Confidence Estimation and Evaluation", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 879–882.

[GlassEtAl95] Glass J., Goddeau D., Hetherington L., McCandless M., Pao C., Phillips M., Polifroni J., Seneff S., and Zue V., "The MIT ATIS System: December 1994 Progress Report", *Proceedings of the ARPA Human Language Technology Workshop*, 1995, pp. 252–256.

[GlassEtAl99] Glass J.R., Hazen T.J., and Hetherington I.L., "Real-Time Telephone-Based Speech Recognition in the Jupiter Domain", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 61–64.

[Gokcen97] Gokcen S., and Gokcen J.M., "A Multilingual Phoneme and Model Set: Toward a Universal Base for Automatic Speech Recognition", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 599–605.

[GoldbergerBurshtein98] Goldberger J., and Burshtein D., "Scaled Random Segmental Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 809–812.

[Gopinath98] Gopinath R.A., "Maximum Likelihood Modeling with Gaussian Distributions for Classification", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 661–664.

[Greenberg98] Greenberg S., "Recognition in a New Key - Towards a Science of Spoken Language", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 1033–1036.

[GreenbergKingsbury97] Greenberg S., and Kingsbury B.E.D., "The Modulation Spectrum: In Pursuit of an Invariant Representation of Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1647–1650.

[GuptaEtAl96] Gupta S.K., Soong F., and Haimi-Cohen R., "High-Accuracy Connected Digit Recognition for Mobile Applications", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 57–60.

[Haeb-Umbach99] Haeb-Umbach R., "Investigations on Inter-Speaker Variability in the Feature Space", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 397–400.

[HainEtAl99] Hain T., Woodland P.C., Niesler T.R., and Whittaker E.W.D., "The 1998 HTK System for Transcription of Conversational Telephone Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 57–60.

[HamakerEtAl98] Hamaker J., Ganapathiraju A., Picone J., and Godfrey J.J., "Advances in Alphadigit Recognition Using Syllables", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 421–424.

[HanazawaEtAl97] Hanazawa K., Minami Y., and Furui S., "An Efficient Search Method for Large-Vocabulary Continuous-Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1787–1790.

[Hansen94] , Hansen J.H.L., "Morphological Constrained Feature Enhancement with Adaptive Cepstral Compensation (MCE-ACC) for Speech Recognition in Noise and Lombard Effect,", *IEEE Transactions on Speech and Acoustic Processing*, 1994, vol. 2, pp. 598–614.

[HataokaEtAl98] Hataoka N., Kokubo H., Obuchi Y., and Amano A., "Development of Robust Speech Recognition Middleware on Microprocessor", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 837–840.

[Hauenstein97] Hauenstein M., "A Computationally Efficient Algorithm for Calculating Loudness Patterns of Narrowband Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1311–1314.

[HermanSukkar97] Herman S.M., and Sukkar R.A., "Variable Threshold Vector Quantization for Reduced Continuous Density Likelihood Computation in Speech Recognition", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 331–338.

[Hermansky90] Hermansky H., "Perceptual Linear Prediction (PLP) Analysis for Speech", *The Journal of the Acoustical Society of America*, 1990, Vol. 87, pp. 1738–1752.

[Hermansky97] Hermansky H., "The Modulation Spectrum in the Automatic Recognition of Speech", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 140–147.

[HermanskyEtAl85] Hermansky H., Hanson B.A., and Wakita H., "Low-Dimensional Representation of Vowels Based on All-Pole Modeling in the Psychophysical Domain", *Speech Communication*, 1985, pp. 181–187.

[HermanskyEtAl91] Hermansky H., Morgan N., Bayya A., and Kohn P., "Compensation for the effect of the Communication Channel in Auditory-like Analysis of Speech (RASTA-PLP)", *Proceedings of the European Conference on Speech Communication and Technology*, 1991, pp. 1367–1371.

[HermanskyEtAl95] Hermansky H., Morgan N., and Hirsch H-G., "Recognition of Speech in Additive and Convolutive Noise Based on RASTA Spectral Processing", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995, pp. 83–86.

[HommaEtAl97] Homma S., Aikawa K., and Sagayama S., "Improved Estimation of Supervision in Unsupervised Speaker Adaptation", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1023–1026.

[Hon92] Hon H.W., *Vocabulary-Independent Speech Recognition: The VOCIND System*, Ph.D. Thesis, CMU-CS-92-108, School of Computer Science, Carnegie Mellon University, 1992.

[HuBarnard97] Hu Z., and Barnard E., "Smoothness Analysis for Trajectory Features", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 979–982.

[HuaJac89] Huang X.D., and Jack M.A., "Semi-Continuous Markov Models for Speech Signals", *Readings in Speech Recognition*, Academic Press, 1989.

[HuangEtAl90] Huang X.D., Ariki Y., Jack M.A., *Hidden Markov Models for Speech Recognition*, Edinburgh University Press, Edinburgh, 1990.

[HuangEtAl96] Huang X.D., Hwang M-Y., Jiang L., and Mahajan M., "Deleted Interpolation and Density Sharing for Continuous Hidden Markov Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 885–888.

[HwangHuang98] Hwang M-Y., and Huang X., "Dynamically Configurable Acoustic Models for Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 669–672.

[HwangWang97] Hwang J-N., and Wang C-J., "Joint Model and Feature Space Optimization for Robust Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 855–858.

[IllinaGong97] Illina I., and Gong Y., "Elimination of Trajectory Folding Phenomenon: HMM, Trajectory Mixture HMM and Mixture Stochastic Trajectory Model", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1395–1398.

[IyerEtAl97] Iyer R., Ostendorf M., and Meteer M., "Analyzing and Predicting Language Model Improvements", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 254–261.

[JavkinGaller97] Javkin H., Galler M., and Niedzielski N., "Enhancement of Esophageal Speech by Injection Noise Rejection", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1207–1210.

[JinEtAl98] Jin H., Matsoukas S., Schwartz R., and Kubala F., "Fast Robust Inverse Transform Speaker Adapted Training using Diagonal Transformations", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 785–788.

[JitsuhiroEtAl98] Jitsuhiro T., Takahashi S., and Aikawa K., "Rejection of Out-of-Vocabulary Words Using Phoneme Confidence Likelihood", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 217–220.

[JohnsonEtAl99] Johnson S.E., Jourlin P., Moore G.L., Sparck Jones K., and Woodland P.C., "The Cambridge University Spoken Document Retrieval System", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 49–52.

[JohnstonEtAl96] Johnston D., Whittaker S.J., and Attwater D.J., "An Overview of Speech Technology for Telecom Services in the United Kingdom", *1996 IEEE Third Workshop - Interactive Voice Technology for Telecommunications Applications*, 1996, pp. 12–15.

[JonesEtAl96] Jones G.J.F., Foote J.T., Jones K.S., and Young S.J., "Robust Talker-Independent Audio Document Retrieval", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 311–314.

[JouvetEtAl91] Jouvet D., Mauuary L., and Monne J., "Automatic Adjustments of the Structure of Markov Models for Speech Recognition Applications", Eurospeech, Sept. 1991, pp. 927–930

[JouvetEtAl99] Jouvet D., Bartkova K., and Mercier G., "Hypothesis Dependent Threshold Setting for Improved Out-of-Vocabulary Data Rejection", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 709–712.

[JunquaEtAl95] Junqua J.C., "An N-best strategy, Dynamic Grammars and Selectively Trained Neural Networks for Real-Time Recognition of Continuously Spelled Names Over the Telephone", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995.

[JunquaGaller96] Junqua J-C., and Galler M., "Performance Evaluation of *SmarTspelL*: A Continuously Spelled Name Recognizer over the Telephone", *1996 IEEE Third Workshop - Interactive Voice Technology for Telecommunications Applications*, pp. 143–146.

[Junqua97] Junqua J.C., "SmarTspeL: A multi-pass recognition system for name retrieval over the telephone", *IEEE Transactions on Speech and Audio Processing*, 1997, March.

[KalaiEtAl99] Kalai A., Chen S., Blum A., and Rosenfeld R., "On-Line Algorithms for Combining Language Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 745–748.

[KanederaEtAL98] Kanedera N., Hermansky H., and Arai T., "On Properties of Modulation Spectrum for Robust Automatic Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 613–616.

[KannanKhudanpur99] Kannan A., and Khudanpur S., "Tree-Structured Models of Parameter Dependence for Rapid Adaptation in Large Vocabulary Conversational Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 769–772.

[KannanOstendorf97] Kannan A., and Ostendorf M., "Adaptation of Polynomial Trajectory Segment Models for Large Vocabulary Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1411–1414.

[Kao98] Kao Y-H., "Minimization of Search Network in Speech Recognition", *The International Conference on Signal Processing Applications and Technology*, 1998.

[KarnZahorian99] Karnjanadecha M., and Zahorian S.A., "Signal Modeling for Isolated Word Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 293–296.

[KarrayMauuary97] Karray L., and Mauuary L., "Improving Speech Detection Robustness for Wireless Speech Recognition", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 428–435.

[Kawahara97] Kawahara H., "Speech Representation and Transformation Using Adaptive Interpolation of Weighted Spectrum: Vocoder Revisited", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1303–1306.

[KawaharaEtAl97] Kawahara T., Lee C-H., and Juang B-H., "Combining Key-Phrase Detection and Subword-based Verification for Flexible Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1159–1162.

[Keller] Keller E., editor, *Fundamentals of Speech Synthesis and Speech Recognition*, John Wiley & Sons, New York NY, 1994.

[KellnerEtAl96] Kellner A., Rueber B., and Seide F., "A Voice-Controlled Automatic Telephone Switchboard and Directory Information System", *1996 IEEE Third Workshop - Interactive Voice Technology for Telecommunications Applications*, 1996, pp. 117–120.

[KellnerEtAl97] Kellner A., Seide F., and Rueber B., "With a Little Help from the Database - Developing Voice-controlled Directory Information Systems", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, 1997, pp. 566–574.

[KempJusek96] Kemp T., and Jusek A., 'Modeling Unknown Words in Spontaneous Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 530–533.

[KennyEtAl98] Kenny O.P., Nelson S.J., Bodenschatz J.S., and McMonagle H.A., "Separation of Non-spontaneous and Spontaneous Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 573–576.

[KimEtAl99] Kim J., Haimi-Cohen R., and Soong F., "Hidden Markov Models with Divergence Based Vector Quantized Variances", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 125–128.

[KimballOstendorf92] Kimball O., Ostendorf M., and BechWati I., "Context Modeling with the Stochastic Segment Model", *IEEE Transactions on Signal Processing*, 40, no. 6, 1992, pp. 1584–1587.

[KirchhoffBilmes99] Kirchhoff K., and Bilmes J.A., "Dynamic Classifier Combination in Hybrid Speech Recognition Systems Using Utterance-Level Confidence Values", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 693–696.

[KirkpatrickEtAl83] Kirkpatrick S., Gelatt C.D., and Vecchi M.P., "Optimization by Simulated Annealing", *Science*, 220, 1983, pp. 671–680.

[Klakow98] Klakow D., "Language Model Optimization by Mapping of Corpora", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 701–704.

[KneserPeters97] Kneser R., and Peters J., "Semantic Clustering for Adaptive Language Modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 779–782.

[KnillYoung96] Knill K.M., and Young S.J., "Fast Implementation Methods for Viterbi-based Word-spotting", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 522–525.

[KooEtAl97] Koo M-W., Lee C-H., and Juang B-H., "A New Hybrid Decoding Algorithm for Speech Recognition and Utterance Verification", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 303–310.

[KooEtAl98] Koo M-W., Lee C-H., and Juang N-H., "A New Decoder Based on a Generalized Confidence Score", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 213–216.

[KorkmazskiyEtAl97] Korkmazskiy F., and Juang B-H., "Discriminative Training of the Pronunciation Networks", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 223–229.

[KuhnEtAl97] Kuhn R., Nowell P., and Drouin C., "Approaches to Phoneme-based Topic Spotting: An Experimental Comparison", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1811–1814.

[KuhnEtAl99] Kuhn R., Nguyen P., Boman R., Niedzielski N., Fincke S., Field K., and Contolini M., "Fast Speaker Adaptation Using A Priori Knowledge", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 749–752.

[LamKas86] Lamel L.F., Kassel R.H., and Seneff S. "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus", *Proceedings of the Speech Recognition Workshop*, (DARPA) 1986, pp. 100–109.

[Laurila97] Laurila K., "Noise Robust Speech Recognition with State Duration Constraints", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 871–874.

[LaurilaEtAl98] Laurila K., Vasilache M., and Viikki O., "A Combination of Discriminative and Maximum Likelihood Techniques for Noise Robust Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 85–88.

[Lee89] Lee K.F., *Automatic Speech Recognition: The Development of the SPHINX System*, Kluwer, Boston MA, 1989.

[Lee97] Lee C-H., "Adaptive Compensation for Robust Speech Recognition", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 357–364.

[LeeEtAl90A] Lee K.F., Hayamizu S., Hon H.W., Huang C., Swartz J., and Weide R., "Allophone Clustering for Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1990, pp. 749–752.

[LeeEtAl90] Lee K.F., Hon H.W., and Reddy R., "An Overview of the SPHINX Speech Recognition System", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 1, January 1990, pp. 35–44.

[LeeEtAl91] Lee C.H., Giachin E., Rabiner L.R., Pieraccini R., and Rosenberg A.E., "Improved Acoustic Modeling for Speaker Independent Large Vocabulary Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1991, pp. 161–164.

[LeeEtAl96] Lee C-H., Soong F.K., and Paliwal K.K., editors, *Automatic Speech and Speaker Recognition - Advanced Topics*, Kluwer Academic Publishers, Norwell MA, 1996.

[LeeHon89] Lee K.F., and Hon H.W., "Speaker-Independent Phone Recognition Using Hidden Markov Models", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 11, November 1989, pp. 1641–1646.

[LeeOShaugh97] Lee C.Z., and O'Shaugnessy D., "Techniques to Achieve Fast Lexical Access", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 303–310.

[LeeRose96] Lee L., and Rose R.C., "Speaker Normalization Using Efficient Frequency Warping Procedures", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 353–356.

[LevLib89] Levinson S.E., Liberman M.Y., Ljolje A., and Miller L.G., "Speaker Independent Phonetic Transcription of Fluent Speech For Large Vocabulary Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* 1989, pp. 441–444.

[LevinPieraccini95] Levin E., and Pieraccini R., "CHRONUS, The Next Generation", *Proceedings of the ARPA Human Language Technology Workshop*, 1995, pp. 269–271.

[LiOShaugnessy96] Li Z., and O'Shaughnessy D., "Using a Transcription Graph for Large Vocabulary Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 121–124.

[Lip82] Liporace L.A., "Maximum Likelihood Estimation for Multivariate Observations of Markov Sources", *IEEE Transactions on Information Theory*, Vol. IT-28, no. 5, September 1982, pp.729–734.

[LippmannCarlson97] Lippmann R.P., and Carlson B.A., "Robust Speech Recognition with Time-varying Filtering, Interruptions, and Noise", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 365–372.

[LjoljeEtAl95] Ljolje A., Riley M., Hindle D., and Pereira F., "The AT&T 60,000 Word Speech-To-Text System", *Eurospeech*, 1995.

[LleidaRose96] Lleida E., and Rose R.C., "Efficient Decoding and Training Procedures for Utterance Verification in Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 507–510.

[LoganMoreno98] Logan B., and Moreno P., "Factorial HMMs for Acoustic Modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 813–816.

[LuoJelinek99] Luo X., and Jelinek F., "Probabilistic Classification of HMM States for Large Vocabulary Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 293–296.

[MaEtAl98] Ma K.W., Zavaliagkos G., and Meteer M., "Sub-Sentence Discourse Models for Conversational Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 693–696.

[MahajanEtAl99] Mahajan M., Beeferman D., and Huang X.D., "Improved Topic-Dependent Language Modeling Using Information Retrieval Techniques", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 541–544.

[Mak84] Makhoul J., and Schwartz R., "Ignorance Based Modeling", *Ignorance Modeling, Invariance, and Variability in Speech Processing*, Perkell J., and Klatt D.H., editors, Erlbaum, 1984.

[MakBocchieri98] Mak B., and Bocchieri E., "Training of Subspace Distribution Clustering Hidden Markov Model", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 673–676.

[MakEtAl97] Mak B., Bocchieri E., and Barnard E., "Stream Derivation and Clustering Scheme for Subspace Distribution Clustering Hidden Markov Model", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 339–346.

[MatsuiHara99] Matsui K., and Hara N., "Enhancement of Esophageal Speech Using Formant Synthesis", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 81–84.

[McCourtEtAl98] McCourt P., Vaseghi S., and Harte N., "Multi-Resolution Cepstral Features for Phoneme Recognition Across Speech Sub-Bands", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 557–560.

[McDonoughByrne99] McDonough J., and Byrne W., "Speaker Adaptation with All-Pass Transforms", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 757–760.

[McDonoughEtAl96] McDonough J., Zavaliagkos G., and Gish H., "An Approach to Speaker Adaptation Based on Analytic Functions", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 721–724.

[McDonoughEtAl97] McDonough J., Anastasakos T., Zavaliagkos G., and Gish H., "Speaker-Adapted Training on the Switchboard Corpus", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1059–1062.

[McKinleyWhipple97] McKinley B.L., and Whipple G.H., "Model Based Speech Pause Detection", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1179–1182.

[MingSmith98] Ming J., and Smith F.J., "Improved Phone Recognition Using Bayesian Triphone Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 409–412.

[MinskyPapert69] Minsky M., and Papert S., *Perceptrons*, MIT Press, Cambridge MA, 1969.

[MirghaforiMorgan98] Mirghafori N. and Morgan N., "Transmissions and Transitions: A Study of Two Common Assumptions in Multi-Band ASR", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 713–716.

[MitchellSetlur99] Mitchell C.D., and Setlur A.R., "Improved Spelling Recognition Using a Tree-Based Fast Lexical Match", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 597–600.

[MohriEtAl98] Mohri M., Riley M., Hindle D., Ljolje A., and Pereira F., "Full Expansion of Context-Dependent Networks in Large Vocabulary Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 665–668.

[MooreEtAl95] Moore R., Appelt D., Dowding J., Gawron J.M., and Moran D., "Combining Linguistic and Statistical Knowledge Sources in Natural-Language Processing for ATIS", *Proceedings of the ARPA Human Language Technology Workshop*, 1995, pp. 261–264.

[MurveitEtAl93] Murveit H., Butzberger J., Digalakis V., and Weintraub M., "Large-Vocabulary Dictation Using SRI's Decipher Speech Recognition System: Progressive Search Techniques", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1993.

[NageshaGillick97] Nagesha V., and Gillick L., "Studies in Transformation-based Adaptation", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1031–1034.

[NaitoEtAl98] Naito M., Deng L., and Sagisaka Y., "Speaker Clustering for Speech Recognition Using the Parameters Characterizing Vocal-tract Dimensions", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 981–984.

[NetiEtAl97] Neti C.V., Roukos S., and Eide E., "Word-based Confidence Measures as a Guide for Stack Search in Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 883–886.

[NetiRoukos97] Neti C., and Roukos S., "Phone-context Specific Gender-dependent Acoustic Models for Continuous Speech Recognition", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 192–198.

[NeyAubert94] Ney H., and Aubert X., "A Word Graph Algorithm for Large Vocabulary, Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1994.

[NeyEtAl97] Ney H., Ortmanns S., and Lindam I., "Extensions to the Word Graph Method for Large Vocabulary Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1791–1794.

[NeyEtAl98] Ney H., Welling L., Ortmanns S., Beulen K., and Wessel F., "The RWTH Large Vocabulary Continuous Speech Recognition System", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 853–856.

[NeyNoll88] Ney H., and Noll A., "Phoneme Modeling Using Continuous Mixture Densities", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1988, pp. 437–440.

[NeyOrtmanns97] Ney H., and Ortmanns S., "Progress in Dynamic Programming Search for LVCSR", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 287–294.

[NguyenEtAl99] Nguyen P., Gelin P., Junqua J.-C., and Chien J.T., "N-Best Based Supervised and Unsupervised Adaptation for Native and Non-native Speakers in Cars", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 173–176.

[NguyenSchwartz99] Nguyen L., and Schwartz R., "Single-Tree Method for Grammar-Directed Search", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 613–616.

[NieslerEtAl98] Niesler T.R., Whittaker E.W.D., and Woodland P.C., "Comparison of Part-of-Speech and Automatically Derived Category-Based Language Models for Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 177–180.

[NieslerWoodland96] Niesler T.R., and Woodland P.C., "A Variable-length Category-based *N*-gram Language Model", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 164–167.

[NieslerWoodland97] Niesler T.R., and Woodland P.C., "Modeling Word-pair Relations in a Category-based Language Model", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 795–798.

[NiyogiEtAl99] Niyogi P., Burges C., and Ramesh P., "Distinctive Feature Detection Using Support Vector Machines", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 425–428.

[NiyogiRamesh98] Niyogi P., and Ramesh P., "Incorporating Voice Onset Time to Improve Letter Recognition Accuracies", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 13–16.

[Norm91] Normandin Y., *Hidden Markov Models, Maximum Mutual Information Estimation, and the Speech Recognition Problem*, Ph.D. Thesis, Department of Electrical Engineering, McGill University, 1991.

[NothEtAl96] Noth E., Demori R., Fischer J., Gebhard A., Harbeck S., Kompe R., Kuhn R., Niemann H., and Mast M., "An Integrated Model of Acoustics and Language Using Semantic Classification Trees", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 419–422.

[O'ShaughnessyTolba99] O'Shaughnessy D., and Tolba H., "Towards a Robust/Fast Continuous Speech Recognition System Using a Voiced-Unvoiced Decision", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 413–416.

[O'Shaughnessy87] O'Shaughnessy D., *Speech Communication*, Addison-Wesley, Reading MA, 1987.

[ObuchiEtAl97] Obuchi Y., Amano A., and Hataoka N., "A Novel Speaker Adaptation Algorithm and its Implementation on a RISC Microprocessor", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 442–449.

[OkawaEtAl98] Okawa S., Bocchieri E., and Potamianos A., "Multi-Band Speech Recognition in Noisy Environments", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 641–644.

[OrtmannsEtAl97] Ortmanns S., Eiden A., Ney H., and Coenen N., "Look-Ahead Techniques for Fast Beam Search", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1783–1786.

[OrtmannsEtAl98] Ortmanns S., Eiden A., and Ney H., "Improved Lexical Tree Search for Large Vocabulary Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 817–820.

[PadmanabhanEtAl96] Padmanabhan M., Bahl L.R., Nahamoo D., and Picheny M.A., "Speaker Clustering and Transformation for Speaker Adaptation for Large Vocabulary Speech Recognition Systems", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 701–704.

[PadmanabhanEtAl97] Padmanabhan M., Jan E.E., Bahl L.R., and Picheny M., "Decision-tree based Feature-Space Quantization for Fast Gaussian Computation", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 325–310.

[PadmanabhanEtAl98] Padmanabhan M., Eide E., Ramabhadran B., Ramaswamy G., and Bahl L.R., "Speech Recognition Performance on a Voicemail Transcription Task", *Proceedings of the IEEE International*

Conference on Acoustics, Speech, and Signal Processing, 1998, pp. 913–916.

[Paul85] Paul D.B., "Training of HMM Recognizers by Simulated Annealing", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1985, pp. 13–16.

[Paul97] Paul D.B., "Extensions to Phone-State Decision-Tree Clustering: Single Tree and Tagged Clustering", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1487–1490.

[PeskinEtAl96] Peskin B., Connolly S., Gillick L., Loew S., McAllaster D., Nagesha V., van Mulbregt P., and Wegmann S., "Improvements in Switchboard Recognition and Topic Identification", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 303–306.

[PeskinEtAl97] Peskin B., Gillick L., Liberman N., Newman M., van Mulbregt P., and Wegmann S., "Progress in Recognizing Conversational Telephone Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1811–1814.

[PeskinEtAl99] Peskin B., Newman M., McAllaster D., Nagesha V., Richards H., Wegmann S., Hunt M., and Gillick L., "Improvements in Recognition of Conversational Telephone Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 53–56.

[PitasVenetsanopoulos90] Pitas I., and Venetsanopoulos, A.N., *Nonlinear Digital Filters*, Kluwer Academic Publishers, Boston, 1990.

[Pols97] Pols L.C.W., "Flexible Human Speech Recognition", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 273–283.

[PolymenakosEtAl98] Polymenakos L., Olsen P., Kanvesky D., Gopinath R.A., Gopalakrishnan P.S., and Chen S., "Transcriptions of Broadcast News - Some Recent Improvements to IBM's LVCSR System", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 901–904.

[PotamianosRose97] Potamianos A., and Rose R.C., "On Combining Frequency Warping and Spectral Shaping in HMM Based Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1275–1278.

[PoveyWoodland99] Povey D., and Woodland P.C., "Frame Discrimination Training of HMMs for Large Vocabulary Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 333–336.

[PressEtAl88] Press W.H, Flannery B.P., Teukolsky S.A., and Vetterling W.T., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1988.

[PyeWoodland97] Pye D., and Woodland P.C., "Experiments in Speaker Normalization and Adaptation for Large Vocabulary Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1047–1050.

[Qi90] Qi Y., "Replacing Tracheoesophageal Voicing Sources Using LPC Synthesis", *Journal of the Acoustical Society of America*, 1990, Vol. 88, pp. 1228–1235.

[QiEtAl95] Qi Y., Weinberg B., and Bi N., "Enhancement of Female Esophageal and Tracheoesophageal Speech", *Journal of the Acoustical Society of America*, 1995, Vol. 98, pp. 2461–2465.

[Rabiner89] Rabiner L.R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech *Proceedings of the IEEE*, 77, no. 2, February 1989, pp. 257–285.

[Rabiner97] Rabiner L.R., "Applications of Speech Recognition in the Area of Telecommunications", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 501–510.

[RabinerEtAl89] Rabiner L.R., Lee C.H., Juang B.H., and Wilpon J.G., "HMM Clustering for Connected Word Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 405–408.

[RabinerJuang93] Rabiner L., and Juang B.H., *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs NJ, 1993.

[Rahim99] Rahim M., "Recognizing Connected Digits in a Natural Spoken Dialog", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 153–156.

[RahimEtAl95] Rahim M.G., Lee C.H., and Juang B.H., "Robust Utterance Verification for Connected Digit Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995, pp. 285–288.

[RahimSaul97] Rahim M., and Saul L., "Minimum Classification Error Factor Analysis (MCE-FA) for Automatic Speech Recognition", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 172–178.

[RamalingamEtAl97] Ramalingam C.S., Netsch L., and Kao Y-H., "Speaker-Independent Name Dialing with Out-of-Vocabulary Rejection", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1475–1478.

[RamalingamEtAl99] Ramalingam C.S., Gong Y., Netsch L.P., Anderson W.W., Godfrey J.J., and Kao Y.-H., "Speaker-Dependent Name Dialing in a Car Environment with Out-of-Vocabulary Rejection", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 165–168.

[RamanRamanujam97] Raman V., and Ramanujam V., "Robustness Issues and Solutions in Speech Recognition Based Telehony Services", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1523–1526.

[RamaswamyGopa98] Ramaswamy G.N., and Gopalakrishnan P.S., "Compression of Acoustic Features for Speech Recognition in Network Environments", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 977–980.

[RaoEtAl98] Rao A., Rose K., and Gersho A., "Deterministically Annealed Design of Speech Recognizers and its Performance on Isolated Letters", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 461–464.

[ReichlChou97] Reichl W., and Chou W., "A Fast Segmental Clustering Approac hto Decision Tree Tying Based Acoustic Modeling", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 185–191.

[ReichlChou98] Reichl W., and Chou W., "Decision Tree State Tying based on Segmental Clustering for Acoustic Modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 801–804.

[ReichlChou99] Reichl W., and Chou W., "A Unified Approach of Incorporating General Features in Decision Tree Based Acoustic Modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 573–576.

[ReinhardNiranjan98] Reinhard K., and Niranjan M., "Parametric Subspace Modeling of Speech Transitions", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 1105–1108.

[ReinhardNiranjan99] Reinhard K., and Niranjan M., "Diphone Multi-Trajectory Subspace Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 1001–1004.

[RenalsHochberg96] Renals S., and Hochberg M., "Efficient Evaluation of the LVCSR Search Space Using the NOWAY Decoder", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 149–152.

[RiccardiEtAl97] Riccardi G., Gorin A.L., Ljolje A., and Riley M., "A Spoken Language System for Automated Call Routing", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1143–1146.

[RichardsEtAl97] Richards H.B., Bridle J.S., Hunt M.J., and Mason J.S., "Vocal Tract Shape Trajectory Estimation Using MLP Analysis-by-Synthesis", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1287–1290.

[RigazioJunquaGaller98] Rigazio L., Junqua J-L., and Galler M., "Multilevel Discriminative Training for Spelled Word Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 489–492.

[Riis98] Riis S.K., "Hidden Neural Networks: Applications to Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 1117–1120.

[Riley91] Riley M., "A Statistical Model for Generating Pronunciation Networks", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1991.

[RivlinEtAl96] Rivlin Z., Cohen M., Abrash V., and Chung T., "A Phone-dependent Confidence Measure for Utterance Verification", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 515–517.

[RobbinsEtAl84] Robbins J., Fisher H.B., Blom E.C., and Singer M.I., "A Comparative Acoustic Study of Normal, Esophageal, and Tracheoesophageal Speech Production", *Speech Hear Res*, 1984, 49:202–210.

[Robinson91] Robinson T., "Several Improvements to a Recurrent Error Propagation Phone Recognition System", *Technical report TINFENG/TR.82 - Cambridge University Engineering Department*, 1991.

[RobinsonChristie98] Robinson T., and Christie J., "Time-first Search for Large Vocabulary Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 829–832.

[RobinsonFallside90] Robinson T., and Fallside F., "The Cambridge Recurrent Error Propagation Network Speech Recognition System", *Computer Speech and Language*, 1990.

[RoeWilpon] Roe D.B., and Wilpon J., editors, *Voice Communication between Humans and Machines*, National Academy Press, Washington D.C., 1994.

[Rose96] Rose R.C., "Keyword detection in conversational utterances using hidden Markov model based continuous speech recognition", *Computer Speech and Language*, 1995, 9, pp. 309–333.

[RoseEtAl95] Rose R.C., Juang B.H., and Lee C.H., "A Training Procedure for Verifying String Hypotheses in Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995, pp. 281–284.

[RoseEtAl97] Rose R., Yao H., Riccardi G., and Wright J., "Integrating Multiple Knowledge Sources for Utterance Verification in a Large Vocabulary Speech Understanding System", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 215–222.

[RoseEtAl98] Rose R.C., Yao H., Riccardi G., and Wright J., "Integration of Utterance Verification with Statistical Language Modeling and Spoken Language Understanding", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 237–240.

[RoseLleida97] Rose R.C., and Lleida E., "Speech Recognition Using Automatically Derived Acoustic Baseforms", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1271–1274.

[RoseRiccardi99] Rose R.C., and Riccardi G., "Modeling Disfluency and Background Events in ASR for a Natural Language Understanding Task", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 341–344.

[Rosenfeld97] Rosenfeld R., "A Whole Sentence Maximum Entropy Language Model", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 230–237.

[RumelhartMcClelland86] Rumelhart D.E., and McClelland J.L., *Parallel Distributed Processing*, MIT Press, Cambridge MA, 1986.

[Russell97] Russell M., "Progress towards Speech Models that Model Speech", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 115–123.

[SagayamaEtAl97] Sagayama S., Yamaguchi Y., Takahashi S., and Takahashi J., "Jacobian Approach to Fast Acoustic Model Adaptation", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 835–838.

[SagayamaEtAl97a] Sagayama S., Yamaguchi Y., and Takahashi S., "Jacobian Adaptation of Noisy Speech Models", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 396–403.

[SamuelssonReichl99] Samuelsson C., and Reichl W., "A Class-Based Language Model for Large Vocabulary Speech Recognition Extracted from Part-of-Speech Statistics", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 537–540.

[SanchisCasabuberta91] Sanchis E., and Casacuberta F., "Learning Structural Models of Subword Units Through Grammatical Inference Techniques", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1991, pp. 189–192.

[SankarEtAl96] Sankar A., Neumeyer L., and Weintraub M., "An Experimental Study of Acoustic Adaptation Algorithms", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 713–716.

[SchaafKemp97] Schaaf T., and Kemp T., "Confidence Measures for Spontaneous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 875–878.

[ScheyttEtAl98] Scheytt P., Geutner P., and Waibel A., "Serbo-Croatian LVCSR on the Dictation and Broadcast News Domain", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 897–900.

[SchluterMacherey98] Schluter R., and Macherey W., "Comparison of Discriminative Training Criteria", *Proceedings of the IEEE International*

Conference on Acoustics, Speech, and Signal Processing, 1998, pp. 493–496.

[SchmidBarnard97] Schmid P., and Barnard E., "Explicit, *N*-Best Formant Features for Vowel Classification", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 991–994.

[SchwartzEtAl85] Schwartz R., Chow Y., Kimball O., Roucos S., Krasner M., and Makhoul J., "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1985, pp. 1205–1208.

[Schwenk99] Schwenk H., "Using Boosting to Improve a Hybrid HMM/Neural Network Speech Recognizer", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 1009–1012.

[SenZue88] Seneff S., and Zue V.W., "Transcription and Alignment of the TIMIT Database", (distributed with the DARPA TIMIT CD-ROM by the National Bureau of Standa April 9, 1988.

[ShinodaLee97] Shinoda K., and Lee C-H., "Structural MAP Speaker Adaptation Using Hierarchical Priors", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 381–387.

[ShinodaLee98] Shinoda K., and Lee C-H., "Unsupervised Adaptation using Structural Bayes Approach", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 793–796.

[SingerOstendorf96] Singer H., and Ostendorf M., "Maximum Likelihood Successive State Splitting", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 601–604.

[SinghEtAl99] Singh R., Raj B., and Stern R.M., "Automatic Clustering and Generation of Contextual Questions for Tied States in Hidden Markov Models", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 117–120.

[SiohanGong96] Siohan O., and Gong Y., "A Semi-continuous Stochastic Trajectory Model for Phoneme-based Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 471–474.

[SiuEtAl99] Siu M., Jonas M., and Gish H., "Using a Large Vocabulary Continuous Speech Recognizer for a Constrained Domain with Limited Training", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 105–108.

[SixtusOrtmanns99] Sixtus A., and Ortmanns S., "High Quality Word Graphs Using Forward-Backward Pruning", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 593–596.

[SmithBates93] Smith G.W., and Bates M., "Voice Activated Automated Telephone Call Routing", *Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications*, March 1993.

[SolomonoffEtAl98] Solomonoff A., Mielke A., Schmidt M., and Gish H., "Clustering Speakers by their Voices", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 753–756.

[Stewart73] Stewart G.W., *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[StolckeOmohundro] Stolcke A., and Omohundro S.M., "Best-first Model Merging for Hidden Markov Model Induction", Tech. Report 94-003, Berkeley.

[StolckeShriberg96] Stolcke A., and Shriberg E., "Statistical Language Modeling for Speech Disfluencies", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 405–408.

[StropeAlwan98] Strope B., and Alwan W., "Robust Word Recognition Using Threaded Spectral Peaks", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 625–628.

[Sukkar98] Sukkar R.A., "Subword-based Minimum Verification Error (SB-MVE) Training for Task Independent Utterance Verification", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 229–232.

[SukkarEtAl96] Sukkar R.A., Setlur A.R., Rahim M.G., and Lee C.H., "Utterance Verification of Keyword Strings Using Word-based Minimum Verification Error (WB-MVE) Training", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 518–521.

[SukkarLee96] Sukkar R.A., and Lee C-H., "Vocabulary Independent Discriminative Utterance Verification for Nonkeyword Rejection in Subword Based Speech Recognition", *IEEE Transactions on Speech and Audio Processing*, Vol. 4, No. 6, 1996, pp. 420–429.

[Sun97] Sun D.X., "Statistical Modeling of Co-Articulation in Continuous Speech Based on Data Driven Interpolation", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1751–1754.

[SuzukiEtAl97] Suzuki Y., Fukumoto F., and Sekiguchi Y., "Domain Identification and Keyword Extraction of Radio News Using Term Weighting", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 543–550.

[TakamiSagayama92] Takami J., and Sagayama S., "A Successive State Splitting Algorithm for Efficient Allophone Modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1992, pp. I573–576.

[TakaraEtAl97] Takara T., Higa K., and Nagayama I., "Isolated Word Recognition Using the HMM Structure Selected by the Genetic Algorithm", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 967–970.

[Tanaka98] Tanaka K., "Next Major Application Systems and Key Techniques in Speech Recognition Technology", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 1057–1060.

[Tanimoto87] Tanimoto S.L., *The Elements of Artificial Intelligence*, Computer Science Press, Rockville Maryland, 1987.

[ThelenEtAl97] Thelen E., Aubert X., and Beyerlein P., "Speaker Adaptation in the Philips System for Large Vocabulary Continuous Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1035–1038.

[Thomson95] Thomson M., "Statistical Modeling of Speech Vector Trajectories", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995.

[Thomson97] Thomson D.L., "Ten Case Studies of the Effect of Field Conditions on Speech Recognition Errors", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pp. 511–518.

[ThomsonChenga98] Thomson D.L., and Chengalvarayan R., "Use of Periodicity and Jitter as Speech Recognition Features", *Proceedings of*

the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1998, pp. 21–24.

[TibrewalaHermansky97] Tibrewala S., and Hermansky H., "Subband Based Recognition of Noisy Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1255–1258.

[TorreEtAl96] de la Torre C., Hernandez-Gomez L., Caminero-Gil F.J., and del Alamo C.M., "On-line Garbage Modeling for Word and Utterance Verification in Natural Numbers Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 845–848.

[TsakalidisEtAl99] Tsakalidis S., Digalakis V., and Neumeyer L., "Efficient Speech Recognition Using Subvector Quantization and Discrete Mixture HMMs", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 569–572.

[TsugeEtAl99] Tsuge S., Fukada T., and Singer H., "Speaker Normalized Spectral Subband Parameters for Noise Robust Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 285–288.

[UmeshEtAl99] Umesh S., Cohen L., and Nelson D., "Fitting the Mel Scale", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 217–220.

[ValtchevEtAl96] Valtchev V., Odell J.J., Woodland P.C., and Young S.J., "Lattice-Based Discriminative Training for Large Vocabulary Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 605–608.

[VaseghiEtAl97] Vaseghi S., Harte N., and Milner B., "Multi-Resolution Phonetic/Segmental Features and Models for HMM-based Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1263–1266.

[Vergin98] Vergin R., "An Algorithm for Robust Signal Modeling in Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 969–972.

[ViikkiEtAl98] Viikki O., Bye D., and Laurila K., "A Recursive Feature Vector Normalization Approach for Robust Speech Recognition in Noise", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 733–736.

[Vite67] Viterbi A.J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Transactions on Information Theory*, April 1967, vol. 13, no. 2.

[WardIssar95] Ward W., and Issar S., "The CMU ATIS System", *Proceedings of the ARPA Human Language Technology Workshop*, 1995, pp. 249–251.

[WardIssar96] Ward W., and Issar S., "A Class Based Language Model for Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 416–418.

[WegmannEtAl96] Wegmann S., McAllaster D., Orloff J., and Peskin B., "Speaker Normalization on Conversational Telephone Speech", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 339–341.

[WegmannEtAl99] Wegmann S., Zhan P., and Gillick L., "Progress in Broadcast News Transcription at Dragon Systems", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 33–36.

[WeibelEtAl89] Weibel A., Hanazawa T., Hinton G., Shikano K., and Lang K., "Phoneme Recognition Using Time-Delay Neural Networks", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1989, vol. 37, pp. 393–404.

[WeinbergBosna70] Weinberg B., and Bosna J.F., "Similarities between Glossopharyngeal Breathing and Injection Methods of Air Intake for Esophageal Epeech", *Speech and Hearing Disorders*, 1970, no. 35: pp. 25–32.

[WeintraubEtAl97] Weintraub M., Beaufays F., Rivlin Z., Konig Y., and Stolcke A., "Neural-Network Based Measures of Confidence for Word Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 887–890.

[WellingEtAl99] Welling L., Kanthak S., and Ney H., "Improved Methods for Vocal Tract Normalization", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 761–764.

[WendemuthEtAl99] Wendemuth A., Rose G., and Dolfing J.G.A., "Advances in Confidence Measures for Large Vocabulary", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 705–708.

[WilponEtAl90] Wilpon J.G., Rabiner L.R., Lee C.H., and Goldman E.R., "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38, No. 11, 1990, pp. 1870–1877.

[Winston77] Winston P.H., *Artificial Intelligence*, Addison-Wesley, Reading MA, 1977.

[Wokurek97] Wokurek W., "Time-Frequency Analysis of the Glottal Opening", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1435–1438.

[WoodlandEtAl96] Woodland P.C., Gales M.J.F., and Pye D., "Improving Environmental Robustness in Large Vocabulary Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 65–68.

[WoodlandEtAl97] Woodland P.C., Gales M.J.F., Pye D., and Young S.J., "Broadcast News Transcription Using HTK", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 719–722.

[WoodlandEtAl98] Woodland P.C., Hain T., Johnson S.E., Niesler T.R., Tuerk A., and Young S.J., "Experiments in Broadcast News Transcription", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 909–912.

[WoszcynaFinke96] Woszcyna M., and Finke M., "Minimizing Search Errors Due to Delayed Bigrams in Real-Time Speech Recognition Systems", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 137–140.

[WrightEtAl95] Wright J.H., Carey M.J., and Parris E.S., "Topic discrimination using higher-order statistical models of spotted keywords", *Computer Speech and Language*, 1995, no. 9, pp. 381–405.

[WuEtAl97] Wu S-L., Shire M.L., Greenberg S., and Morgan N., "Integrating Syllable Boundary Information into Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 987–990.

[WuEtAl98] Wu S-L., Kingsbury B.E.D., Morgan N, and Greenberg S., "Incorporating Information from Syllable-length Time Scales into Automatic Speech Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 721–724.

[YamamotoEtAl94] Yamamoto S., Takeda K., Inoue N., Kuroiwa S., and Naito M., "A Voice-activated Telephone Exchange System and its Field Trial", *1994 IEEE Second Workshop - Interactive Voice Technology for Telecommunications Applications*, 1994, pp. 21–26.

[YamamotoSagisaka99] Yamamoto H., and Sagisaka Y., "Multi-Class Composite N-Gram based on Connection Direction", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 533–536.

[Young89] Young R.K., *Wavelet Theory and its Applications*, Kluwer, Boston MA, 1989.

[Young92] Young S.J., "The General Use of Tying in Phoneme-based HMM Speech Recognizers", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1992, pp. 569–572.

[YoungWoodland94] Young S.J., and Woodland P.C., "State Clustering in hidden Markov model-based continuous speech recognition", *Computer Speech and Language*, 1994, Vol. 8, No. 4, pp. 369–383.

[YuEtAl98] Yu H., Clark C., Malkin R., and Waibel A., "Experiments in Automatic Meeting Transcription Using JRTK", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 921–924.

[ZavaliagkosEtAl96] Zavaliagkos G., Schwartz R., and McDonough J., "Maximum A Posteriori Adaptation for Large Scale HMM Recognizers", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 725–728.

[ZavaliagkosEtAl98] Zavaliagkos G., McDonough J., Miller R., El-Jaroudi A., Billa J., Richardson F., Ma K., Siu M., and Gish H., "The BBN BYBLOS 1997 Large Vocabulary Conversational Speech Recognition System", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, pp. 905–908.

[Zeljkovic96] Zeljkovic I., "Decoding Optimal State Sequence with Smooth State Likelihoods", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, pp. 129–132.

[ZeppenfeldEtAl97] Zeppenfeld T., Finke M., Ries K., Westphal M., and Waibel A., "Recognition of Conversational Telephone Speech Using the JANUS Speech Engine", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1811–1814.

[ZhanWestphal97] Zhan P., and Westphal M., "Speaker Normalization Based on Frequency Warping", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1039–1042.