An Overview of the SPHINX Speech Recognition System

KAI-FU LEE, MEMBER, IEEE, HSIAO-WUEN HON, AND RAJ REDDY, FELLOW, IEEE

Abstract—**Speaker independence, continuous speech, and large vocabularies pose three of the greatest challenges in automatic speech recognition. Previously, accurate speech recognizers avoided dealing simultaneously with all three problems. This paper describes SPHINX, a system that demonstrates the feasibility of accurate, large-vocabulary speaker-independent, continuous speech recognition.**

SPHINX is based on discrete hidden Markov models (HMM's) with LPC-derived parameters. To provide speaker independence, we added knowledge to these HMM's in several ways: multiple codebooks of fixed-width parameters, and an enhanced recognizer with carefully designed models and word duration modeling. To deal with coarticulation in continuous speech, yet still adequately represent a large vocabulary, we introduce two new subword speech units—function-word-dependent phone models and generalized triphone models. With grammars of perplexity 997, 60, and 20, SPHINX attained word accuracies of 71, 94, and 96 percent on a 997-word task.

I. Introduction

CONSIDERABLE progress has been made in speech recognition in the past 15 years. Many successful systems [1]-[7] have emerged. Each of these systems has attained very impressive accuracy. However, they owe their success to one or more of the constraints they impose. This paper describe SPHINX, a system that tries to overcome three of these constraints: 1) speaker dependence, 2) isolated words, and 3) small vocabulary.[1]

Speaker independence has been viewed as the most difficult constraint to overcome. This is because most parametric representations of speech are highly speaker dependent, and a set of reference patterns suitable for one speaker may perform poorly for another speaker. Researchers have found that errors increased by 300-500 percent when a speaker-dependent system is trained and tested in speaker-independent mode [8], [9]. Because of these difficulties, most speech recognition systems are speaker dependent. In other words, they require a speaker to "train" the system before reasonable performance can be expected. This training phase typically requires several hundred sentences. While speaker-trained systems are useful for some applications, they are inconvenient, less robust, more wasteful, and simply unusable for some applications. Speaker-independent systems must train on less appropriate training data. However, many more data can be acquired, which may compensate for the less appropriate training material.

Continuous speech recognition is significantly more difficult than isolated word recognition. Its complexity is a result of three innate properties of continuous speech. First, word boundaries are difficult to locate. Second, *coarticulatory effects* are much stronger in continuous speech, causing the same sound to appear differently in various contexts. Third, *content words* (nouns, verbs, adjectives, etc.) are often emphasized, while *function words* (articles, prepositions, pronouns, short verbs, etc.) are poorly articulated. Error rates increase drastically from isolated-word to continuous speech. For example, Bahl *et al.* [10] reported a 280 percent error rate increase from isolated-word to continuous speech recognition. However, in spite of these problems and degradations, we believe that it is important to work on continuous speech research. Only with continuous speech can we achieve the desired speed and naturalness of man-machine communications.

Large vocabulary typically implies a vocabulary of about 1000 words or more. As vocabulary size increases, so does the number of confusable words. Also, larger vocabularies require the use of *subword models*, because it is difficult to train whole word models. Unfortunately, subword units usually lead to degraded performance because they cannot capture coarticulatory (interunit) effects as well as word models can. Error rate increased by 200-1000 percent in several studies [11]-[13]. In spite of these problems, large vocabulary systems are still needed for many versatile applications, such as dictation, dialog systems, and speech translation systems.

In this paper, we describe SPHINX, a large-vocabulary speaker-independent, continuous speech recognition system. SPHINX employs discrete hidden Markov models (HMM's) with LPC-derived parameters. To deal with speaker independence, we added knowledge to these HMM's in several ways. We represented additional knowledge through the use of multiple vector quantized codebooks. We also enhanced the recognizer with carefully designed models and word duration modeling. To

Manuscript received July 5, 1988; revised March 22, 1989. This work was supported in part by a National Science Foundation Graduate Fellowship, and by Defense Advanced Research Projects Agency Contract N00039-85-C-0163. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the Defense Advanced Research Projects Agency, or the U.S. Government.

The authors are with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

IEEE Log Number 8931685.

[1] There are many other constraints that SPHINX does impose: simple language model, benign environment, cooperative speakers, etc.

0096-3518/90/0100-0035$01.00 © 1990 IEEE

deal with coarticulation in continuous speech, yet adequately represent a large vocabulary, we introduced two new speech units—function-word-dependent phone models and generalized triphone models. With these techniques, SPHINX achieved speaker-independent word recognition accuracies of 71, 94, and 96 percent on the 997-word DARPA resource management task [14] with grammars of perplexity 997, 60, and 20.

In this paper, we first describe the task and database used for evaluating SPHINX in the following section. Section III then describes a baseline implementation of SPHINX. Enhancements to SPHINX using additional human knowledge and improved subword models are described in Sections IV and V. Section VI summarizes the results with SPHINX, and Section VII concludes with some final remarks. A full description of the SPHINX System can be found in [15] and [16].

II. Task and Database

A. The Resource Management Task

SPHINX was evaluated on the DARPA *resource management* task [14]. This task, containing a vocabulary of 997 words, was designed for database query of naval resources. As such, there are a large number of long words, such as *Apalachicola*, *Chattahoochee*, and *ECG041*. These words are relatively easy to recognize. On the other hand, it also contains many confusable pairs, such as *what/what's, what/was, the/a, four/fourth, are/were, any/many,* etc. Also, there are many function words (such as *a, and, of, the, to*), which are articulated very poorly and are hard to recognize or even locate. In particular, *the* and *a* are the most frequent words, but are optional according to the grammar.

The original grammar designed for the resource management task was a finite state grammar. This grammar had a perplexity of only about 9, which was too simple. Instead, we used three more difficult grammars with SPHINX: 1) null grammar (perplexity 997), where any word can follow any other word, 2) word-pair grammar (perplexity 60), a simple grammar that specifies a list of words that can legally follow any given word, and 3) bigram grammar (perplexity 20), a word-pair grammar that uses word-category transitions probabilities estimated from the grammar. It should be noted that the training and testing sentences were generated from the finite state grammar, which may reduce acoustic confusability [17].

B. The TIRM Database

Texas Instruments supplied Carnegie Mellon with a large speech database for the resource management task described in the previous section. The TIRM database contains 80 "training" speakers, 40 "development test" speakers, and 40 "evaluation speakers." At the time of this writing, only the 80 training speakers and the 40 development test speakers are available. Of these speakers, 85 are male and 35 are female, with each speaker reading 40 sentences generated by the sentence pattern grammar.

These sentences were recorded using a Sennheiser HMD-414-6, close-talking noise-cancelling headset-boom microphone in a sound-treated room. All speakers were untrained and instructed to read a list of sentences in a natural continuous fashion. The speech was sampled at 20 kHz at TI, downsampled to 16 kHz at the National Institute of Standards and Technology and saved on magnetic tapes.

In this study, all 80 training speakers, as well as 25 of the development test speakers, were used as training material. This gave us a total of 4200 training sentences. The remaining 15 development test speakers were set aside as testing speakers. Ten sentences were taken from each speaker, for a total of 150 test sentences.

III. The Baseline SPHINX System

To establish a performance benchmark using standard HMM techniques on the resource management task, we began with a baseline HMM system. This system uses standard HMM techniques employed by many other systems [18]-[20]. We will show that, using these techniques alone, we can already attain reasonable, albeit mediocre, accuracies.

A. Speech Processing

The speech is sampled at 16 kHz, and preemphasized with a filter whose transform function is $1-0.97z^{-1}$. The waveform is then blocked into frames. Each frame spans 20 ms, or 320 speech samples. Consecutive frames overlap by 10 ms, or 160 speech samples. Each frame is multiplied by a Hamming window with a width of 20 ms and applied every 10 ms.

From these smoothed speech samples, we computed the LPC coefficients using the autocorrelation method [21]. LPC analysis was performed with order 14. Finally, a set of 12 LPC-derived cepstral coefficients was computed from the LPC coefficients. This representation is very similar to that used by Shikano *et al.* [22] and Rabiner *et al.* [23].

The 12 LPC cepstrum coefficients for each frame were then vector quantized into one of 256 prototype vectors. These vectors were generated by a variant of the Linde–Buzo–Gray algorithm [24], [22] using Euclidean distance. We used 150 00 frames of nonoverlapped 20-ms coefficients extracted from 4000 sentences to generate the 256-vector codebook.

B. Phonetic Hidden Markov Models

Hidden Markov Models (HMM) were first described by Baum [25]. Shortly afterwards, they were independently extended to automatic speech recognition by Baker [26] and Jelinek [27]. However, only in the past few years have HMM's become the predominant approach to speech recognition.

HMM's are parametric models particularly suitable for describing speech events. The success of HMM's is largely due to the forward–backward reestimation algorithm [19], which is a special case of the EM algorithm [25]. Every iteration of the algorithm modifies the parameters to increase the probability of the training data until a local maximum has been reached.

Because the resource management task is a large-vocabulary one, we cannot adequately train a model for each word. Thus, we have chosen to use phonetic HMM's, where each HMM represents a phone. There are a total of 45 phones, each characterized by

- $\{s\}$—a set of states including an initial state S_I and a final state S_F,
- $\{a_{ij}\}$—a set of transitions where a_{ij} is the probability of taking a transition from state i to state j,
- $\{b_{ij}(k)\}$-the output probability matrix: the probability of emitting symbol k when taking a transition from state i to state j, k corresponds to one of the 256 VQ codes.

Each phonetic HMM has the topology shown in Fig. 1. The three self-loops model three parts of a phone, and the lower transitions explicitly model durations of one, two, or three frames. Instead of assigning a unique output pdf to each transition, each phone is assigned three distributions, representing the beginning, middle, and end of the phone. Each of these three distributions is shared by several transitions. This model is almost identical to that used by IBM [28].

C. Training

To initialize our phone model parameters, we used hand-segmented and hand-labeled segments from 2240 TIMIT [29] sentences. We ran one iteration of forward–backward on these hand-labeled phone segments, and produced a model for each phone. This set of 45 phone models was used to initialize the parameters in the actual training.

After this initialization, we ran the forward–backward algorithm on the resource management (TIRM) training sentences. For each of the 4200 sentences, we created a sentence model from word models, which were in turn concatenated from phone models. To determine the phonetic spelling of a word, we used a pronunciation dictionary adopted from the baseform of the ANGEL System [30], where each word is mapped to a single linear sequence of phones. Then, to create a sentence model from word models, we accounted for possible between-word silences by inserting a mandatory silence model at the beginning and at the end of the sentence. Between-word silences were also allowed, but were optional. This sentence model represents the *expected pronunciation* of the sentence. It was trained against the actual input speech using the forward–backward algorithm [19].

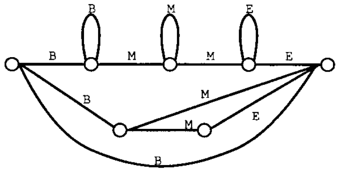

Fig. 1. The phone HMM used in baseline SPHINX. The label on a transition represents the output pdf to which the transition is tied.

Two iterations of forward–backward training were then run. Most other HMM systems run more iterations, but we found that with our appropriate initialization, two iterations were sufficient. The trained transition probabilities were used directly in recognition. The output probabilities, however, were smoothed with a uniform distribution to avoid probabilities that were too small.

The SPHINX recognition search is a standard time-synchronous Viterbi beam search [19], [20]. The search processes input speech time synchronously, completely updating all accessible states for a time frame $t - 1$ before moving on to frame t. The update for time t consists of two stages. First, for each within-word transition between states s_{from} and S_{to}, if $P(s_{from}, t - 1)$. $P(transition) \cdot P(output)$ is greater than $P(s_{to}, t)$, then $P(s_{to}, t)$ is updated. Second, for the final state of every word, all legal word successors are tried, using $P(transition)$ derived from the language model.

In the Viterbi beam search, a hypothesis is pruned if its log probability is less than that of the best hypothesis by more than a preset threshold. We found it is possible to prune 80–90 percent of the hypotheses without any loss in accuracy. After the search is completed, a backtrace is performed to recover the best path.

D. Results

The results with the baseline SPHINX system, using 15 new speakers with 10 sentences each for evaluation, are shown in Table I. To determine the recognition accuracy, we first align the recognized word string against the correct word string using a string match algorithm supplied by the National Institute of Standards and Technology [31]. This alignment determines *WordsCorrect*, *Substitutions*, *Deletions*, *Insertions*. Finally, *PercentCorrect* and *WordAccuracy* are computed by

$$\textit{Percent Correct} \\ = 100 \cdot \frac{\textit{Words Correct}}{\textit{Correct Length}} \qquad (1)$$

$$\textit{Word Accuracy} \\ = 100 \cdot \frac{\textit{Correct Length—Subs—Dels—Ins}}{\textit{Correct Length}}. \qquad (2)$$

Confusions between homonyms (such as *ship's* and *ships*, or *two* and *too*) are not counted for the null language model, and are counted for the word pair and the bigram language model.

The results of this system are mediocre at best. Since

TABLE I
BASELINE SPHINX RESULTS, EVALUATED ON 150 SENTENCES FROM 15 SPEAKERS

Grammar	Perplexity	Percent Correct	Word Accuracy
None	997	31.1%	25.8%
Word-Pair	60	61.8%	58.1%
Bigram	20	76.1%	74.8%

the bigram grammar already imposes tight constraints, we concluded that our baseline system was inadequate for any realistic large-vocabulary applications. In the subsequent sections, we describe our steps to improve the baseline SPHINX by incorporating knowledge and contextual modeling.

IV. ADDING KNOWLEDGE TO SPHINX

A. Fixed-Width Speech Parameters

The easiest way to add knowledge to HMM's is to introduce additional fixed-width parameters, or parameters than can be computed for every fixed-size frame. All we have to do is to devise a way of incorporating these parameters into the output pdf of the HMM's. In this section, we consider several types of frame-based parameters, and discuss possible ways of integrating them.

1) Bilinear Transform on the Cepstrum Coefficients: The human ear's ability to discriminate between frequencies is approximated by a logarithmic function of the frequency, or a *bark scale* [32]. Furthermore, Davis and Mermelstein [33] have shown these logarithmically scaled coefficients yield superior recognition accuracy compared to linearly scaled ones. Therefore, there is strong motivation for transforming the LPC cepstrum coefficients into a mel-frequency scale.

Shikano [34] reported significant improvement from using a *bilinear transform* [35] on the LPC cepstral coefficients. Bilinear transform is a technique that transforms a linear frequency axis into a warped one using the all-pass filter

$$z_{new}^{-1} = \frac{(z^{-1} - a)}{(1 - az^{-1})}, \quad (-1 < a < 1) \qquad (3)$$

$$\omega_{new} = \omega + 2 \tan^{-1}\left(\frac{a \sin \omega}{1 - a \cos \omega}\right) \qquad (4)$$

where ω is the sampling frequency expressed by the normalized angular frequency, ω_{new} is the converted frequency, and a is a frequency warping parameter. A positive a converts the frequency axis into a low-frequency weighted one. When a takes on values between 0.4 and 0.8, the frequency warping by a bilinear transform is comparable to that of the mel or Bark scales. In this work, we use a value of 0.6 for a.

2) Differenced Cepstrum Coefficients: Temporal changes in the spectra play an important role in human perception [36]. This is particularly true for speaker-independent recognition, where formant slopes are more reliable than absolute formant locations. Thus, it would be desirable to incorporate "slope" measurements into recognizers. Moreover, since HMM's assume each frame is independent of the past, it would be desirable to broaden the scope of a frame.

We use a simple slope measure, *differenced LPC cepstrum coefficients* [34]. The difference coefficients for frame n are the difference between the coefficients of frame $n + \delta$ and $n - \delta$. In our current implementation, a differenced coefficient is computed every frame, with $\delta = 2$ frames, giving a 40 ms difference. In a preliminary experiment, we found this measure to be as good as the *regression coefficients* used in [37] and [7].

3) Power and Differenced Power: Although LPC-based parameters perform well in speech recognition, they do not contain sufficient information about power. For example, coefficients in silence or noise regions are not very meaningful. Therefore, it is desirable to incorporate power into our recognizer. Rabiner *et al.* [23] obtained significant improvement by adding power into the distance metric in vector quantization, and Shikano [34] reported similar results. Finally, in a detailed study of prosody in speech recognition, Waibel [38] found power to be the most important prosodic cue.

Since raw power may vary widely from speaker to speaker, we normalized power by subtracting the maximum power value in the sentence from each power value in the sentence. In our real-time system, we used an automatic gain control algorithm with a 250-ms look-ahead to predict the maximum power in a sentence.

Another important source of information is *differenced power*, which is computed the same way as *differenced LPC cepstrum coefficients*. Differenced power provides information about relative changes in amplitude or loudness. Indeed, our preliminary experiments indicated that differenced power is more useful than power.

4) Integrating Fixed-Width Parameters in Multiple Codebooks: There are many ways to integrate the above coefficients into the framework of a discrete HMM recognizer. We considered several possibilities [15], and decided to use *multiple-codebook integration* [39]. Using this technique, coefficients are divided into sets, and each set is quantized into a separate codebook. We created three codebooks, each with 256 codes. These codebooks were generated from 1) bilinear-transformed LPC cepstrum coefficients, 2) differenced bilinear-transformed LPC cepstrum coefficients, and 3) a weighted combination of power and differenced power.

For each frame of speech, not one but several VQ codes are used to replace the input vector. Since each input frame is no longer a single symbol, but rather a vector of symbols, the discrete HMM algorithms must be modified to produce multiple symbols at each time frame. By assuming that the multiple output observations are independent, the output probability of emitting multiple symbols can then be computed as the product of the probability of producing each symbol.

The multiple-codebook approach has a distinct advantage over single-codebook approaches—namely, reduced quantization error. If too many features are used in VQ, the distortion will be very large, which means the observed vectors will match their corresponding prototype vectors poorly. Multiple codebooks reduce the distortion by partitioning the feature space into several smaller subspaces. Table II clearly illustrates this point with the comparison of one-codebook distortion and three-codebook distortion.

TABLE II
QUANTIZATION ERROR OF A SINGLE CODEBOOK VERSUS THE TOTAL QUANTIZATION ERROR IN THREE CODEBOOKS

Codebook Size	1-codebook distortion	3-codebook distortion
2	2.42	1.86
4	1.94	1.12
8	1.45	0.81
16	1.19	0.61
32	1.00	0.48
64	0.83	0.39
128	0.72	0.31
256	0.61	0.25

Another advantage of multiple codebooks is the large increase in the dynamic range and precision of the resulting parameters. With three codebooks, there are 256^3 possible parameter combinations using just 256 × 3 parameters. With such an increase in precision comes the ability to make finer distinctions.

However, the independence assumption with multiple codebooks is inaccurate. Also, more memory and time are needed with multiple codebooks. But we felt that these disadvantages were well compensated by the advantages.

B. Lexical/Phonological Improvements

Our next set of improvements involved the modification of the set of phones and the pronunciation dictionary. These changes lead to more accurate assumptions about how words are articulated, without changing our assumption that each word has a single pronunciation.

The first step we took was to replace the baseform pronunciation with the most likely pronunciation. For example, the first vowel of the word *delete* will appear as /iy/ in most dictionaries, but it is actually pronounced as /ih/ most of the time. This correction process modified about 40 percent of all the baseforms.

With our linear representation of pronunciation, it is difficult to model the deletions of phonetic events. For example, the first /d/ of the word *did* is always released, while the last /d/ may be unreleased. Also, closures before stops are optional. We model these two types of deletions implicitly in the HMM parameters. We created separate models for the released stops and optional stops. We also merged closure-stop pairs as a single phone. These changes enabled the modeling of deletions within linear HMM's.

Although the English phonemes are well defined, there are actually many frequently used sounds that are not phonemic. For example, stop-fricative pairs such as /ks/, /ps/, /ts/, /bz/, /dz/, or /gz/ are actually quite different from the concatenated phoneme pairs. They appear more like different affricates. Thus, it is sensible to model them as special phones. In this study, we only model /ts/ in this fashion due to the lack of training data for the other nonphonemic affricates.

In order to improve the appropriateness of the word pronunciation dictionary, a small set of rules was created to 1) modify closure-stop pairs into optional compound phones when appropriate, 2) modify /t/'s and /d/'s into /dx/ when appropriate, 3) reduce nasal /t/'s when appropriate, and 4) perform other mappings such as /t s/ to /ts/.

Finally, there is the issue of what HMM topology is optimal for phones in general, and what topology is optimal for each phone. We found that although the choice of model was not critical for continuous speech recognition, the model shown in Fig. 1 led to the best results. In addition, we experimented with different ways of labeling the transitions, i.e., which output pdf should be tied to each transition. Each phone was assigned an appropriate set of tied transitions.

The improvements in this section led to the set of phones enumerated in Table III. These improvements have increased the number of phones from 45 to 48. Table IV shows a section of our final phonetic pronunciation dictionary.

C. Word Duration Modeling

HMM's model duration of events with transition probabilities, which lead to a geometric distribution for the duration of state residence, for states with self-loops:

$$P_i(d) = (1 - a_{ii})\, a_{ii}^d \qquad (5)$$

where $P_i(d)$ is the probability of taking the self-loop at state i for exactly d times. Several researchers have argued that this is an inadequate distribution for speech events, and proposed alternatives for duration modeling [40], [41], [7].

We incorporated word duration into SPHINX as a part of the Viterbi search. The duration of a word is modeled by a univariate Gaussian distribution, with the mean and variance estimated from a supervised Viterbi segmentation of the training set. By precomputing the duration score for various durations, this duration model has essentially no overhead.

D. Results

We have presented various strategies for adding knowledge to SPHINX. The results of these strategies are shown in Table V. The version abbreviations are defined in Table VI.

Consistent with earlier results [33], [34], we found that bilinear transformed coefficients improved the recognition rates. An even greater improvement came from the use of differential coefficients, power, and differenced power in three separate codebooks. Next, we enhanced the dictionary and the phone set—a step that led to an appreciable improvement.

TABLE III
LIST OF THE IMPROVED SET OF PHONES IN SPHINX

Phone	Example	Phone	Example	Phone	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/	*button*	/th/	*thief*
/ah/	*but*	/m/	*mom*	/s/	*sis*
/uw/	*boot*	/n/	*non*	/sh/	*shoe*
/uh/	*book*	/ng/	*sing*	/hh/	*hay*
/ao/	*bought*	/ch/	*church*	/sil/	(silence)
/aa/	*cot*	/jh/	*judge*	/dd/	*deleted*
/ey/	*bait*	/dh/	*they*	/pd/	*ship*
/ay/	*bite*	/b/	*bob*	/td/	*set*
/oy/	*boy*	/d/	*dad*	/kd/	*comic*
/aw/	*bough*	/g/	*gag*	/dx/	*butter*
/ow/	*boat*	/p/	*pop*	/ts/	*its*

TABLE IV
A SECTION OF THE SPHINX DICTIONARY WITH WORD, ORIGINAL BASEFORM, AND THE PRONUNCIATION AFTER RULE APPLICATION

Word	Baseform	After rules
ADDED	/ae d ix d/	/ae dx ix dd/
ADDING	/ae d ix ng/	/ae dx ix ng/
AFFECT	/ax f eh k t/	/ax f eh k td/
AFTER	/ae f t er/	/ae f t er/
AGAIN	/ax g eh n/	/ax g eh n/
AJAX	/ey jh ae k s/	/ey jh ae k s/
ALASKA	/ax l ae s k ax/	/ax l ae s k ax/
ALERT	/ax l er t/	/ax l er td/
ALERTS	/ax l er t s/	/ax l er ts/

TABLE V
THE SPHINX RESULTS WITH KNOWLEDGE ENHANCEMENTS. RESULTS SHOWN ARE PERCENT-CORRECT (WORD-ACCURACY)

Version	No grammar	Word Pair	Bigram
Baseline	31.1% (25.8%)	61.8% (58.1%)	76.1% (74.8%)
Bilinear Trans.	34.2% (28.6%)	63.1% (59.4%)	78.5% (76.0%)
4F3C	45.6% (40.1%)	83.3% (81.1%)	88.8% (87.9%)
Phonology	50.0% (45.3%)	86.8% (84.4%)	91.2% (90.6%)
Duration	55.1% (49.6%)	85.7% (83.8%)	91.4% (90.6%)

TABLE VI
THE DEFINITION OF THE VERSION ABBREVIATIONS USED IN TABLE V

Version	Description
Baseline	The version in Table I.
Bilinear Trans.	After adding bilinear transform.
4F3C	After adding four feature sets and three codebooks.
Phonology	After all the dictionary and phonological improvements, plus implicit insertion/deletion modeling.
Duration	After integration of word duration probabilities into the Viterbi Search.

Finally, the addition of durational information significantly improved SPHINX's accuracy when no grammar was used, but was not helpful with a grammar. With no grammar, the recognizer must consider many word hypotheses, and word duration modeling can filter out many hypotheses with implausible word durations. On the other hand, when a grammar is used, much more constraint is applied, sharply decreasing the utility of duration. Therefore, in subsequent versions, duration modeling is used only without grammar.

V. CONTEXT MODELING IN SPHINX

Given that we will use hidden Markov models to model speech, one important question is: what unit of speech should an HMM represent? In the previous sections, we have used phones as the fundamental unit of speech. An even more natural unit is words. In this section, we will discuss the strengths and weaknesses of word and phone models, as well as a number of other units proposed by earlier work. Then, we shall propose two new units that will substantially improve the performance of speaker-independent continuous speech recognizers. Finally, we will present comparative results of different variations of these units.

A. Previously Proposed Units of Speech

Words are the most natural units of speech because they are exactly what we want to recognize. Word models are able to capture within-word contextual effects, so by modeling words as units, phonological variations can be assimilated. Therefore, when there are sufficient data, word models will usually yield the best performance. However, using word models in large-vocabulary recognition introduces several grave problems. Since training data cannot be shared between words, each word has to be trained individually. For a large-vocabulary task, this imposes too great a demand for training data and memory. Also, for many tasks, it would be convenient to provide the user with the option of adding new words to the vocabulary. If word models were used, the user would have to produce many repetitions of the word, which would be extremely inconvenient. Therefore, while word models are natural and model contexts well, because of the lack of sharing across words, they are not practical for large-vocabulary speech recognition.

In order to improve trainability, some subword unit has to be used. The most commonly used subword units are the phones of English. The implementation of SPHINX we have described thus far is based on phone models. With only about 50 phones in the English language, they can be sufficiently trained with just a few hundred sentences. We have seen that the earlier implementations of SPHINX yielded reasonably accurate results. However, studies [42], [13] have shown that well-trained word models outperform well-trained phone models. This is because phone models assume a phone in any context is equivalent to the same phone in any other context. However, phones are not produced independently, because our articulators cannot move instantaneously from one position to another. Thus, the realization of a phone is strongly affected by its immediate neighboring phones. Another problem with using phone models is that phones in function words, such as *a*, *the*, *in*, *me*, are often articulated poorly, and are not representative instances of the phones. Thus, while word models lack generality, phone models overgeneralize.

Word-dependent phones [12] are a compromise between word modeling and phone modeling. The parameters of a word-dependent phone model depend on the word in which the phone occurs. Like word models, word-dependent phone models can model word-dependent, phonological variations, but they also require considerable training and storage. However, with word-dependent phones, if a word has not been observed frequently, its parameters can be interpolated (or averaged) with those of context-independent phone models. This obviates the need of observing every word in training, and facilitates the addition of new words.

Another alternative—context-dependent phones [20], [12]—is similar to word-dependent phones; instead of modeling phone-in-word, they model phone-in-context. The most commonly used context-dependent model is the triphone model. A *triphone* model is a phone-size model that takes into consideration the left and the right neighboring phones. Triphone modeling is powerful because it models the most important coarticulatory effects, and is much more sensitive than phone modeling. However, the large number of triphones causes them to be poorly trained, in spite of some robustness provided by interpolating with phones. Moreover, some phonetic contexts are quite similar, and triphones cannot take advantage of that.

B. Function-Word Dependent Phones

Function words are typically prepositions, conjunctions, pronouns, articles, and short verbs, such as *the*, *a*, *in*, *are*. Function words are particularly problematic in continuous speech recognition because they are typically unstressed. Moreover, the phones in function words are distorted in many ways. They may be shortened, omitted, or seriously affected by neighboring contexts. Since these effects are specific to the individual function words, explicit modeling of phones in these function words should lead to a much better representation. Function words have caused considerable problems in SPHINX. Function words take up only 4 percent of the vocabulary, or about 30 percent if weighed by frequency, yet they are accountable for almost 50 percent of the errors.

In view of the above analysis, we propose a new speech unit: *function-word-dependent phones*. Function-word-dependent phones are the same as word-dependent phones, except they are only used for function words. This strategy improves the modeling of the most difficult subset of words. Because function words occur frequently in any large-vocabulary task, function-word-dependent phones are readily trainable.

We selected a set of 42 function words (shown in Table VII), for which we felt there were significant word-dependent coarticulatory effects, as well as adequate training data. A few of these words are not usually considered function words, but were appropriate for this task.

TABLE VII
THE LIST OF 42 FUNCTION WORDS THAT SPHINX MODELS SEPARATELY

A	ALL	AND	ANY	ARE	AT	BE
BEEN	BY	DID	FIND	FOR	FROM	GET
GIVE	HAS	HAVE	HOW	IN	IS	IT
LIST	MANY	MORE	OF	ON	ONE	OR
SHOW	THAN	THAT	THE	THEIR	TO	USE
WAS	WERE	WHAT	WHY	WILL	WITH	WOULD

C. Generalized Triphones

Although triphones model the most important coarticulatory effects, they are sparsely trained and consume substantial memory. We now describe a technique to deal with these problems by combining similar triphones. This approach is justified by the fact that some phones have the same effect on neighboring phones [15]. By merging similar triphones, we both improve the trainability and reduce the memory usage.

We created *generalized triphones* by merging contexts with an agglomerative clustering procedure [43].

1) Generate an HMM for every triphone context.
2) Create clusters of triphones, with each cluster consisting of one triphone initially.
3) Find the *most similar* pair of clusters that represents the same phone, and merge them together.
4) For each pair of clusters, consider moving every element from one to the other.
   i) Move the element if the resulting configuration is an improvement.
   ii) Repeat until no such moves are left.
5) Until some convergence criterion is met, go to step 2.

To determine the similarity between two models, we use the following distance metric:

$$D(a, b) = \frac{\left(\prod_i \left(P_a(i)\right)^{N_a(i)}\right) \cdot \left(\prod_i \left(P_b(i)\right)^{N_b(i)}\right)}{\prod_i \left(P_m(i)\right)^{N_m(i)}} \tag{6}$$

where $D(a, b)$ is the distance between two models of the same phone in context a and b. $P_a(i)$ is the output probability of codeword i in model a, and $N_a(i)$ is the forward-backward count of codeword i in model a. m is the merged model obtained by adding N_a and N_b. In measuring the distance between the two models, we only consider the output probabilities and ignore the transition probabilities, which are of secondary importance.

Equation (6) measures the ratio between the probability that the individual distributions generated the training data and the probability that the combined distribution generated the training data. This ratio is consistent with the maximum-likelihood criterion used in the forward-backward algorithm. This distance metric is equivalent to, and was motivated by, entropy clustering used by [44] and [28].

This context generalization algorithm provides the ideal means for finding the equilibrium between trainability and sensitivity. Given a fixed amount of training data, it is possible to find the largest number of trainable detailed models. Armed with this technique, we could attack any problem and find the "right" number of models that are as sensitive and trainable as possible.

D. Smoothing Detailed Models

While the detailed models introduced in the previous sections are more accurate models of acoustics–phonetics, they are less robust because many output probabilities will be zeros, which can be disastrous to recognition. We could intelligently replace the zeros with nonzero probabilities by combining these detailed models with other more robust ones. For example, we could combine the function-word-dependent phone models or the generalized triphone models with the robust context-independent phone models.

An ideal solution for weighting different estimates of the same event is *deleted interpolated estimation* [45]. Deleted interpolation weighs each distribution according to its ability to predict unseen data. By equating these weights to transition probabilities on parallel transitions, the interpolation problem is transformed into an HMM problem, and the weights are learned by the forward–backward algorithm.

In our implementation for training detailed (function-word-dependent and generalized triphone) models, we first initialized the detailed models with the general (context-independent) models. Two iterations of the normal forward–backward algorithm were run using detailed modeling. During the last iteration, we divided the data into two blocks, and maintained separate output and transition counts for each block. After the end of the last iteration, 100 iterations of deleted interpolation were run to combine:

- a detailed model (function-word-dependent or generalized triphone),
- a general model (context-independent phone models—the counts for a general model are the sum of the counts in all the detailed models that correspond to the general model),
- a uniform distribution.

Thus, this procedure not only combined detailed (but less robust) models with robust (but less detailed) models, but also smoothed the distribution using the uniform distribution. The summary of the entire training procedure is illustrated in Fig. 2.

E. Results

Table VIII shows that the direct modeling of phones in function words substantially reduced errors. Table IX gives the number of errors (substitutions + deletions + insertions) made by SPHINX (context-independent models, no grammar) with and without the use of function-word-dependent phone models. With function-word-dependent phone modeling, function word errors are cut by 27 percent, which accounts for almost all of the improvement from 45.3 to 53.4 percent accuracy.

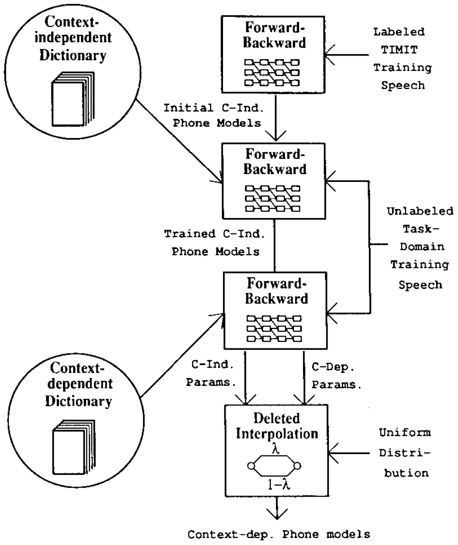
Context-independent Dictionary
Forward-Backward
Labeled TIMIT Training Speech
Initial C-Ind. Phone Models
Forward-Backward
Unlabeled Task-Domain Training Speech
Trained C-Ind. Phone Models
Forward-Backward
Context-dependent Dictionary
C-Ind. Params.
C-Dep. Params.
Deleted Interpolation
λ
1-λ
Uniform Distribution
Context-dep. Phone models

Fig. 2. The training procedure in SPHINX.

TABLE VIII
IMPROVEMENT FROM FUNCTION-WORD-DEPENDENT PHONE MODELING AND GENERALIZED TRIPHONE MODELING. RESULTS SHOWN ARE PERCENT-CORRECT (WORD-ACCURACY)

Version	Models	No grammar	Word pair	Bigram
Context-ind.	48	55.1% (49.6%)	86.8% (84.4%)	91.2% (90.6%)
+Fnwd-dep.	153	62.9% (57.0%)	90.6% (87.9%)	93.8% (93.0%)
+Gen. Triphones	1076	74.2% (70.6%)	94.7% (93.7%)	96.2% (95.8%)

TABLE IX
NUMBER OF FUNCTION WORD ERRORS AND NONFUNCTION-WORD ERRORS WITH AND WITHOUT FUNCTION-WORD-DEPENDENT PHONE MODELING. CONTEXT-INDEPENDENT MODELS WERE USED WITHOUT GRAMMAR

Model Type	Function Word Errors	Other Errors
Context-ind.	357	350
CI+fnwd-dep.	261	334

As indicated in Table VIII, generalized triphone modeling led to another substantial improvement. We ran the agglomerative clustering algorithm to reduce 2381 triphones to 1000 generalized triphones. Combined with function-word-dependent phones, there were a total of 1076 models.

More detailed descriptions and results on contextual modeling can be found in [15] and [46].

VI. SUMMARY OF RESULTS

Fig. 3 shows improvements from all versions of SPHINX described in this paper. The six versions in Fig. 3 correspond to the following descriptions with incremental improvements:

1) the baseline system, which uses only LPC cepstral parameters in one codebook;

2) the addition of differenced LPC cepstral coefficients, power, and differenced power in one codebook;

3) all four feature sets were used in three separate

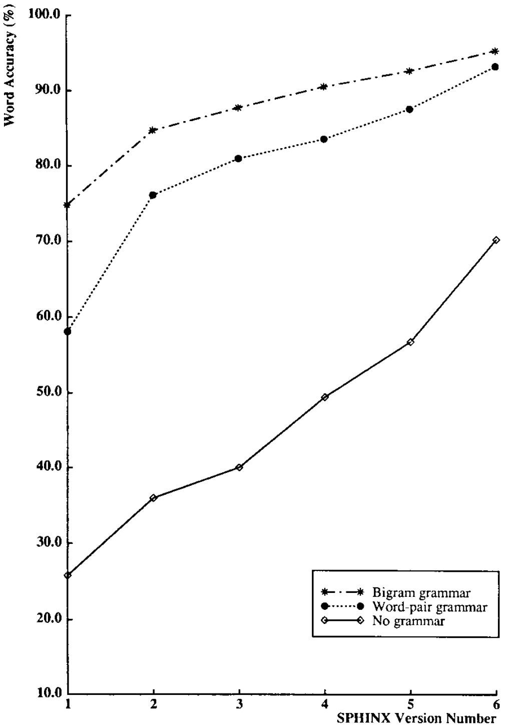

Fig. 3. Results of five versions of SPHINX.

codebooks (this version was reported in [47], the first description of the SPHINX System;

4) tuning of phone models and the pronunciation dictionary, and the use of word duration modeling;

5) function word dependent phone modeling (this version was reported in [48]); and

6) generalized triphone modeling (this version was reported in [15] and [49].

Table X shows the word accuracy, gender, and geographical distribution of the 15 testing speakers. Although the performance appears to vary from speaker to speaker, this variability is not predictable from the speaker's gender or dialect.

VII. Conclusion

We have described SPHINX—a hidden Markov model-based system for large-vocabulary speaker-independent continuous speech recognition. On the one hand, HMM's perform better with detailed models. On the other hand, HMM's need considerable training. This need is accentuated in large-vocabulary speaker-independence, and discrete HMM's. However, given a fixed amount of training, model specificity and model trainability pose two incompatible goals. More specificity usually reduces trainability, and increased trainability usually results in over generality.

Thus, our work can be viewed as finding an equilibrium between specificity and trainability. To improve trainability, we used one of the largest speaker-independent speech databases. To facilitate sharing between models, we used deleted interpolation to combine robust models with detailed ones. By combining poorly trained (context-dependent, generalized context, function-word-dependent speaker-dependent) models with well-trained (context-independent speaker-independent, uniform) models, we improved trainability through sharing.

TABLE X
SPHINX Word Accuracy by Speakers. "Moved" Means that the Speaker Grew Up in More than One Region. Results Shown are Word Accuracy

Initials	Gender	Dialect	No Grammar	Word Pair	Bigram
bcg	F	Moved	61.7%	91.9%	97.7%
sah	F	New Eng.	61.4%	91.0%	94.4%
ljd	F	North Mid.	67.3%	93.7%	98.2%
lmk	F	South	71.3%	96.6%	96.6%
awf	F	South	73.9%	94.4%	95.5%
dpk	M	New Eng.	65.5%	90.2%	93.9%
dab	M	New Eng.	71.3%	95.5%	98.5%
dlc	M	North Mid.	92.6%	100.0%	100.0%
gwt	M	Northern	83.2%	96.4%	97.6%
ctm	M	Northern	72.7%	89.3%	92.9%
jfc	M	NYC	61.2%	93.4%	88.9%
sjk	M	NYC	80.3%	95.1%	96.3%
ctt	M	South	73.6%	94.3%	98.9%
bth	M	Western	69.8%	97.7%	96.6%
jfr	M	Western	62.0%	88.1%	92.4%

To improve specificity, we used multiple codebooks of various LPC-derived features, and integrated external knowledge sources into the system. We also improved the phone set to include multiple representations of some phones, and introduced the use of function-word-dependent phone modeling and generalized triphone modeling.

Through these techniques we have demonstrated that large-vocabulary speaker-independent continuous speech recognition is feasible. We believe that with a powerful learning paradigm, the performance of a system can always be improved with more training data, subject to our ability to make the models more sophisticated. The sophisticated modeling techniques introduced in this paper reduced the error rate of our baseline system by as much as 85 percent, resulting in accuracies of 71, 94, and 96 percent for a 997-word vocabulary with grammars of perplexity 997, 60, and 20.

Acknowledgment

The authors wish to thank the members of the Carnegie Mellon Speech Group. In particular, we would like to acknowledge Hwang, R. Bisiani, and J. Polifroni. We would also like to thank P. Brown and R. Schwartz for helpful discussions, suggestions, and critique. Finally, we thank R. Taylor for reading drafts of this paper.

References

[1] B. T. Lowerre, "The HARPY speech recognition system," Ph.D. dissertation, Comput. Sci. Dep., Carnegie Mellon Univ., Apr. 1976.

[2] J. G. Wilpon, L. R. Rabiner, and A. Bergh, "Speaker-independent isolated word recognition using a 129-word airline vocabulary," *J. Acoust. Soc. Amer.* vol. 72, no. 2, pp. 390-396, Aug. 1982.

[3] R. A. Cole, R. M. Stern, M. S. Phillips, S. M. Brill, P. Specker, and A. P. Pilant, "Feature-based speaker independent recognition of English letters," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Oct. 1983.

[4] F. Jelinek *et al.*, "A real-time, isolated-word, speech recognition system for dictation transcription," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Mar. 1985.

[5] D. B. Paul, R. P. Lippmann, Y. Chen, and C. Weinstein, "Robust HMM-based techniques for recognition of speech produced under stress and in noise," *Speech Tech.*, Apr. 1986.

[6] Y. L. Chow, M. O. Dunham, O. A. Kimball, M. A. Krasner, G. F. Kubala, J. Makhoul, S. Roucos, and R. M. Schwartz, "BYBLOS: The BBN continuous speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1987, pp. 89–92.

[7] L. R. Rabiner, J. G. Wilpon, and F. K. Soong, "High performance connected digit recognition using hidden Markov models," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1988.

[8] S. E. Levinson, A. E. Rosenberg, and J. L. Flanagan, "Evaluation of a word recognition system using syntax analysis," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1977.

[9] B. T. Lowerre, "Dynamic speaker adaptation in the Harpy speech recognition system," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1977.

[10] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Speech recognition of a natural text read as isolated words," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1981.

[11] A. E. Rosenberg, L. R. Rabiner, J. Wilpon, and D. Kahn, "Demisyllable-based isolated word recognition system," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 713–726, June 1983.

[12] Y. L. Chow, R. Schwartz, S. Roucos, O. Kimball, P. Price, F. Kubala, M. Dunham, M. Krasner, and J. Makhoul, "The role of word-dependent coarticulatory effects in a phoneme-based speech recognition system," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1986.

[13] D. B. Paul and E. A. Martin, "Speaker stress-resistant continuous speech recognition," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1988.

[14] P. J. Price, W. Fisher, J. Bernstein, and D. Pallett, "A database for continuous speech recognition in a 1000-word domain," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1988.

[15] K. F. Lee, "Large-vocabulary speaker-independent continuous speech recognition: The SPHINX system," Ph.D. dissertation, Comput. Sci. Dep., Carnegie Mellon Univ., Apr. 1988.

[16] ——, *Automatic Speech Recognition: The Development of the SPHINX System.* Boston, MA: Kluwer Academic, 1989.

[17] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Recognition results with several experimental acoustic processors," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1979.

[18] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition," *Bell Syst. Tech. J.*, vol. 62, no. 4, pp. 1075–1105, Apr. 1983.

[19] L. R. Bahl, F. Jelinek, and R. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 179–190, Mar. 1983.

[20] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul, "Context-dependent modeling for acoustic-phonetic recognition of continuous speech," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1985.

[21] J. D. Markel, and A. H. Gray, *Linear Prediction of Speech.* Berlin: Springer-Verlag, 1976.

[22] K. Shikano, K. Lee, and D. R. Reddy, "Speaker adaptation through vector quantization," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1986.

[23] L. R. Rabiner, K. C. Pan, F. K. Soong, "On the performance of isolated word speech recognizers using vector quantization and temporal energy contours," *AT&T Bell Lab. Tech. J.*, vol. 63, no. 7, pp. 1245–1260, Sept. 1984.

[24] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol COM-28, pp. 84–95, Jan. 1980.

[25] L. E. Baum, "An inequality and associated maximization technique in statistical estimation of probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972.

[26] J. K. Baker, "The DRAGON system—An overview," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 24–29, Feb. 1975.

[27] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, vol. 64, pp. 532–556, Apr. 1976.

[28] P. Brown, "The acoustic-modeling problem in automatic speech recognition," Ph.D. dissertation, Comput. Sci. Dep., Carnegie Mellon Univ., May 1987.

[29] W. M. Fisher, V. Zue, J. Bernstein, and D. Pallett, "An acoustic-phonetic data base," presented at the 113th Meet. Acoust. Soc. Amer., May 1987.

[30] A. Rudnicky, L. Baumeister, K. DeGraaf, and E. Lehmann, "The lexical access component of the CMU continuous speech recognition system," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1987.

[31] D. Pallett, "Test procedures for the March 1987 DARPA benchmark tests," in *Proc. DARPA Speech Recog. Workshop*, Mar. 1987, pp. 75–78.

[32] E. Zwicker, "Subdivision of the audible frequency range into critical bands (Frequenzgruppen)," *J. Acoust. Soc. Amer.*, vol. 33, p. 248, Feb. 1961.

[33] S. B. Davis and P. Mermelstein, "Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357–366, Aug. 1980.

[34] K. Shikano, "Evaluation of LPC spectral matching measures for phonetic unit recognition," Tech. Rep., Comput. Sci. Dep., Carnegie Mellon Univ., May 1985.

[35] A. V. Oppenheim and D. H. Johnson, "Discrete representation of signals," *Proc. IEEE*, vol. 60, pp 681–691, June 1972.

[36] G. Ruske, "Auditory perception and its application to computer analysis of speech," in *Computer Analysis and Perception*, Auditory Signals, Vol. II, C. Y. Suen and R. De Mori, Eds. Boca Raton, FL: CRC Press, 1982.

[37] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 52–59, Feb. 1986.

[38] A. H. Waibel, "Prosody and speech recognition," Ph.D. dissertation, Comput. Sci. Dep., Carnegie Mellon Univ., Oct. 1986.

[39] V. N. Gupta, M. Lennig, and P. Mermelstein, "Integration of acoustic information in a large vocabulary word recognizer," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1987, pp. 697–700.

[40] M. J. Russel and R. K. Moore, "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1985, pp. 5–8.

[41] S. E. Levinson, "Continuously variable duration hidden Markov models for automatic speech recognition," *Comput. Speech Language*, pp. 29–45, 1986.

[42] L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer, "Acoustic Markov models used in the Tangora speech recognition system," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1988.

[43] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[44] J. M. Lucassen and R. L. Mercer, "An information theoretic approach to the automatic determination of Phonemic baseforms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1984.

[45] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal, Eds. Amsterdam, The Netherlands: North-Holland, 1980, pp. 381–397.

[46] K. F. Lee, "Context-dependent phonetic hidden Markov models for continuous speech recognition," submitted to the *IEEE Trans. Acoust., Speech, Signal Processing.*

[47] "Towards speaker-independent continuous speech recognition," presented at the 1987 NATO ASI Speech Recogn. Dialog Understanding, July 1987.

[48] K. F. Lee and H. W. Hon, "Large-vocabulary speaker-independent continuous speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1988.

[49] K. F. Lee, "On large-vocabulary speaker-independent fcontinuous speech recognition," *J. Euro. Assoc. Signal Processing (Speech Communications)*, no. 7, pp. 375–379, Dec. 1988.

Kai-Fu Lee (S'85-M'88) was born in Taipei, Taiwan, in 1961. He received the A.B. degree (summa cum laude) in computer science from Columbia University, New York, NY, in 1983, and the Ph.D. degree in computer science from Carnegie Mellon University, Pittsburgh, PA, in 1988.

Since May 1988 he has been a Research Computer Scientist at Carnegie Mellon, where he currently directs the speech recognition effort within the speech group. His current research interests include automatic speech recognition, spoken language systems, artificial intelligence, and neural networks.

Dr. Lee is a member of Phi Beta Kappa, Sigma Xi, the Acoustical Society of America, and the American Association of Artificial Intelligence.

Hsiao-Wuen Hon was born on May 31, 1963. He received the B.S. degree in electrical engineering from National Taiwan University in 1985.

Since 1986 he has been a Ph.D. student in the Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA, where he is involved in speech research. From 1985 to 1986 he was a full-time Teaching Assistant at the Department of Computer Science and Information Engineering, National Taiwan University. His research interests include speech recognition, artificial intelligence, neural networks, pattern recognition, stochastical modeling, and signal processing.

Raj Reddy (F'83) is University Professor of Computer Science and Robotics, and Director of the Robotics Institute at Carnegie Mellon University. His current research activities involve the study of artificial intelligence, including speech, vision, and robotics; man/machine communication; applications specific computer architectures; and rapid prototyping. Prior to joining Carnegie Mellon's Department of Computer Science in 1969, he was an Assistant Professor of Computer Science at Stanford University. He also served as an Applied Science Representative for International Business Machines Corporation (IBM) in Sydney, Australia. Currently he is the Chairman for the DARPA Information Science and Technology (ISAT) Study Group; and a member of the Academic Advisory Panel for the Technology Transfer Intelligence Committee (TTIC) and the Computer Science and Technology Board of the National Research Council.

Dr. Reddy is a Fellow of the Acoustical Society of America; member of the National Academy of Engineering; and President of the American Association for Artificial Intelligence (1987–1989). He was presented the Legion of Honor, France's highest honor, by President Mitterrand of France in 1984.