Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition

KAI-FU LEE, MEMBER, IEEE

Abstract—**The effectiveness of context-dependent phone modeling for speaker-dependent continuous speech recognition has recently been demonstrated. In this study, we apply context-dependent phone models to speaker-independent continuous speech recognition, and show that they are equally effective in this domain. In addition to evaluating several previously proposed context-dependent models, we also introduce two new context-dependent phonetic units: 1) function-word-dependent phone models, which focus on the most difficult subvocabulary, and 2) generalized triphones, which combine similar triphones together based on an information-theoretic measure. The subword clustering procedure used for generalized triphones can find the optimal number of models given a fixed amount of training data. We demonstrate that context-dependent modeling reduces the error rate by as much as 60%.**

I. Introduction

THE most successful small-vocabulary speech recognition systems have been based on hidden Markov modeling (HMM) of *words* [2], [3]. However, as the vocabulary size increases, word modeling becomes more difficult because it is unreasonable to expect the many repetitions of each word needed to train the word HMM. Instead, some *subword units* must be used. Two important criteria for good subword units are: 1) consistency—different instances of a unit have similar characteristics, and 2) trainability—sufficient training samples exist to create a robust model. In the 1970's researchers faced the dilemma of choosing larger subword units (syllables, demisyllables, diphones) that are consistent but difficult to train or choosing smaller units (phones, phonemes) that are trainable but inconsistent. The introduction of context-dependent phones [4], [5] opened a new chapter in subword modeling. Context-dependent phones are consistent units. Even though there may be a large number of context-dependent phones because they are phone-like units, they can be interpolated with phones to achieve reasonable trainability and performance.

In this paper, we describe the application of context-dependent phone modeling to large-vocabulary speaker-independent continuous speech recognition. In addition to experiments with previously used subword models, we introduce two new types of context-dependent phonetic units: *function-word-dependent phones* and *generalized triphones*.

Function-word-dependent phone models explicitly model phones in function words, such as *in*, *a*, *the*, *and*. They focus on the most confusable subvocabulary, and can be well-trained because function words occur frequently. Function-word-dependent phone models can be interpolated with context-independent phone models to avoid insufficient training.

Generalized triphone models evolved from the *triphone* model [5]–[7]. A triphone is a phone that takes into consideration its left and right phonetic contexts, which cause the greatest variations in the realization of a phone. Triphone models are typically poorly trained because there are many of them. Although interpolation with better trained models makes them usable, they are still undertrained and wasteful. In view of this, we introduce another new unit, generalized triphones. Generalized triphones exploit the fact that many phonetic contexts are very similar. Prior to training, triphones are clustered into generalized triphones using an information-theoretic measure. This not only results in substantially fewer context-dependent units, but also provides more training data for each model. This technique provides a way of finding the "right" number of models for any task. It can also be extended to clustering other types of phonetic models.

To combine detailed models (function-word-dependent phone models and generalized triphone models) with robust ones (context-independent phone models), we use deleted interpolation [8]. Deleted interpolation is an elegant EM (estimate-maximize) algorithm that estimates the weight of the models based on how well each model predicts unseen data.

These techniques were applied to the DARPA 997-word speaker-independent continuous speech resource management task [9]. With phonetic models, we attained word accuracies of 49.6, 84.4, and 90.6% for three different

Manuscript received July 18, 1988; revised June 12, 1989. This work was supported in part by a National Science Foundation Graduate Fellowship and in part by the Defense Advanced Research Projects Agency under Contract N00039-85-C-0163.

The author is with the School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213.

IEEE Log Number 8934005.

0096-3518/90/0400-0599$01.00 © 1990 IEEE

grammars. With function-word-dependent phone models and generalized triphone models, recognition rates improved to 70.6, 93.7, and 95.8%. This represents error rate reductions of 41, 60, and 55%, respectively.

The organization of this paper is as follows. Section II provides a literature review of subword models. Section III describes the SPHINX speech recognition system, the resource management task, and database. In Section IV, we describe our implementation and results of several previously proposed subword models. In Sections V and VI, we introduce function-word-dependent phone modeling and generalized triphone models. Section VII describes directions of future work. Finally, Section VIII contains the conclusions of this paper.

II. Literature Review

Good subword units should be *consistent* and *trainable*. Consistency means that different instances of the same subword model have similar characteristics. Consistency is important because it improves the discrimination between different subword units, which governs the accuracy of speech recognition systems. Trainability means that each speech unit has been trained on sufficiently many examples. Trainability is important because our speech models require a considerable amount of training data. In this section, we will describe and evaluate a number of units using these two criteria.

A. Words

Words are the most natural units of speech because they are exactly what we want to recognize. Also, word models are able to capture within-word contextual effects. For example, the phone /t/ in *ten* is as expected, the phone /t/ in *thirty* is usually flapped, and the phone /t/ in *twenty* may be deleted. By modeling words as units, these phonological variations can be assimilated. Therefore, when there is sufficient data, word models will usually yield the best performance. This is demonstrated by the success of several recent small-vocabulary word-based recognizers [2], [3].

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. Thus, many examples of each word are needed for adequate training. But for a large-vocabulary task, this imposes too great a demand for training data. This problem is difficult for speaker-independent systems, and even more difficult for speaker-dependent ones. Another problem is that memory usage grows linearly with the number of words since there is no sharing between words. Finally, for many tasks, it would be convenient to provide the user with the option of adding new words to the vocabulary. If word models are used, the user would have to produce many repetitions of each new word, which is 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.

B. Phones

In order to allow sharing across words, some subword unit has to be used. The subword units most familiar to us are the phones of English. Since there are only about 50 phones in English, they can be sufficiently trained with just a few hundred sentences. Therefore, unlike word models, there is no training problem with phone models. However, phone models assume that a phone in any context is equivalent to the same phone in any other context. Yet, this is far from the truth. Although we may try to say each word as a concatenated sequence of phones, these 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. Fig. 1 illustrates coarticulatory effects on the phone /t/ in four different contexts. Another problem with using phone models is that phones in function words, such as *a*, *the*, *in*, and *me*, are often articulated poorly, and are not representative instances of the phones.

Bahl *et al.* [10] showed that word-based DTW performed significantly better than phone-based HMM for speaker-dependent recognition. Paul [7] also demonstrated that word-based HMM's resulted in a 50% error rate reduction from phone-based HMM's. These results demonstrate that while word models lack generality, phone models overgeneralize.

C. Multiphone Units

One way to model coarticulatory effects is to use larger units of speech. Examples of this include syllables [11] or demisyllables [12]. These units encompass the phone clusters that contain the most severe contextual effects. However, while the central portions of these units have no contextual dependencies, the beginning and ending portions are still susceptible to some contextual effects.

A more serious problem is the large number of these units. For example, there are over 20 000 syllables and over 1000 demisyllables in English. Although this may be a reduction from word models in a very large vocabulary, there are still too many parameters to reliably estimate when different units cannot share the same training data. Finally, experiments [12] showed that a demisyllable-based recognizer performed substantially worse than a word-based recognizer.

D. Explicit Transition Modeling

Since transitions in and out of phones are poorly modeled by phone models, one possible solution is to model these transitional regions explicitly. For example, diphones [13], [14] model pairs of phones without the use of stationary phones. Another approach is to use stationary phone models and insert transition models [15].

Transition modeling suffers from the same problem as multiphone units. Instead of N phones, there are N^2 phone

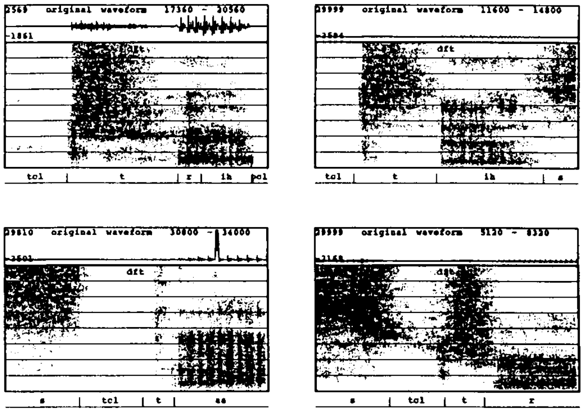

Fig. 1. The waveforms and spectrograms for the phoneme /t/ in four different contexts: part of /t r/, left of /ih/, part of /s t/, and part of /s t r/. It is clear that the realization of /t/ is highly dependent on context, and that a context-independent /t/ model is inadequate.

transitions. Like multiphone units, these units cannot easily share training data. Therefore, transition models also results in too many parameters to estimate when there is no sharing.

E. Word-Dependent Phones

Word-dependent phones [16] 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 that of context-independent phone models. This obviates the need of observing every word in training, and facilitates the addition of new words.

Word-dependent phone modeling was proposed by Chow *et al.* [16]. In that study, word-dependent phone models were interpolated with context-independent phone models using empirically determined weights. Word-dependent phone modeling yielding a 10% error rate, while word models had a 14% error rate, and phone models had a 24% error rate. Thus, word-dependent phone models actually outperformed word models because some word models were poorly trained, while the corresponding word-dependent phone models were reasonably trained through interpolation.

F. Triphones (Context-Dependent Phones)

Context-dependent phone models are similar to word-dependent phone models, except that instead of modeling phone-in-word, they model phone-in-context. A context usually refers to the immediate left and/or right neighboring phones. A *left-context dependent phone* is dependent on the left context, while a *right-context dependent phone* is dependent on the right context. A *triphone* model takes into consideration both the left and the right neighboring phones; if two phones have the same identity but different left or right context, they are considered different triphones. Triphone models are usually poorly trained because there are many triphones. But since triphone models are specific phone models, they can be interpolated with better trained but less appropriate context-independent phone models.

Bahl *et al.* [4] first proposed context-dependent phonetic models. Schwartz *et al.* [5] at BBN were the first to publish comparative results of triphone modeling. In that study and later studies [1], [16], triphone models were interpolated with right-context-dependent models (phone models that are dependent on the right context), left-context-dependent models, and context-independent models. Each pdf in each model was given a different weight according to appropriateness (for example, left-context models have greater weights for leftmost pdf) and amount of training (for example, if a triphone has been observed many times, its weight will dominate). This weight matrix was tuned by hand. For both phone and word recognition, modeling phone-in-context reduced the error rate by about 50%.

Triphone modeling is powerful because it models the most important coarticulatory effect, and is much more consistent and detailed than phone modeling. However, triphone models have two problems. The first problem is memory wastage. When a triphone is observed once, a model is created for it, and with a large number of triphones, the memory used could be substantial. Also, many triphone models are poorly trained, and they do not take advantage of the fact that many triphones are similar.

G. Summary of Previous Units

In the preceding sections, we have evaluated several previously proposed units of speech. We emphasized two important properties for speech units, namely, consistency and trainability. If we have infinite training data, consistency is the only property of interest. But because our training data are not only finite, but often limited, trainability becomes an important issue. Trainability can be achieved either by using very general units (such as phones) at the cost of inconsistency or by sharing among units.

Table I evaluates the appropriateness of the units we described for large-vocabulary recognition using these two criteria. Words are consistent units, but they are not easily trainable. Phones are not consistent, but are easily trainable. Multiphone units and transition units are consistent, but are not easily trainable because there are no good means of sharing. Word-dependent phones and context-dependent phones are consistent, and can be trained because there exist means of sharing. Therefore, both are very appealing units. Our only criticism is that consistency is achieved with too fine a level of detail. More generalization can lead to fewer models and a higher level of trainability.

TABLE I
EVALUATION OF PREVIOUSLY PROPOSED UNITS OF SPEECH TO LARGE VOCABULARY RECOGNITION

Unit	Consistency	Trainability
Word model	Yes	No
Phone model	No	Yes
Multi-phone model	Yes	Difficult
Transition model	Yes	Difficult
Word-dependent phone model	Yes	Through Sharing
Context-dependent phone model	Yes	Through Sharing

III. THE SPHINX SYSTEM

In this section, we describe the SPHINX speech recognition system, on which we experiment with various context-dependent phonetic models. We describe the speech processing, training, and recognition algorithms used in SPHINX. More details about the SPHINX system can be found in [17], [18].

A. Speech Processing

The speech is sampled at 16 kHz, and preemphasized using the transfer function $1 - 0.97z^{-1}$. A Hamming window with a width of 20 ms is applied every 10 ms. 14th-order LPC analysis is implemented using the autocorrelation method [19]. Next, a set of 12 LPC cepstral coefficients is computed from the LPC coefficients. Finally, these LPC cepstral coefficients are transformed to a mel scale using a bilinear transform [20].

In addition to the LPC cepstral coefficients, we also compute differenced LPC cepstral coefficients, power, and differenced power for each frame. The temporal difference for frame t is the difference between frame $t - 2$ and $t + 2$ or a 40 ms difference. In speaker-independent recognition, the use of differential information and power information is extremely important [21], [22].

These coefficients are then vector quantized using three VQ codebooks, each with 256 prototype vectors, using

1) 12 LPC cepstral coefficients
2) 12 differenced LPC cepstral coefficients
3) power and differenced power.

We found that multiple codebooks reduce the VQ distortion by reducing the dimensions of the parameter space. Multiple codebooks were first used by Gupta *et al.* [23].

B. Phonetic Hidden Markov Models

Hidden Markov models (HMM) [24]–[26] are parametric models particularly suitable for describing speech events. The success of HMM's is largely due to the forward–backward reestimation algorithm [26], which is special case of the EM algorithm [27]. Every iteration of the algorithm modifies the parameters to increase the probability of the training data until a local maximum has been reached.

Because our emphasis is on large-vocabulary recognition, we cannot train a model for each word. Thus, we have chosen to use phonetic HMM's where each HMM represents a phone. A phonetic HMM is 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.

We used an inventory of 48 phones (shown in Table II) as subword units. In the initial version, each phone is modeled independent of context. The topology of our model is shown in Fig. 2. The label of a transition designates the output pdf of that transition. The unlabeled lower transitions are assigned different output pdf's for different phones. This model is almost identical to that used by IBM [28].

Because we use three VQ codebooks, our discrete HMM's must produce three VQ codewords at each time frame. We assume that the three output pdf's are independent, which allows us to compute the output probability as the product of the probabilities of emitting the three VQ codewords.

C. HMM Training

The SPHINX training procedure operates in two stages. In the first stage, 48 context-independent phonetic models are trained. After initializing with HMM's trained for phone recognition [22], we run the forward–backward algorithm on a resource management database [9]. This database consists of 4200 sentences (105 speakers with 40 sentences each). For each sentence, a sentence model is created by concatenating the phone models that represent the words in the sentence, with optional silence models between words. Since the entire sentence model is trained, there is no need for hand segmentation or labeling. After two iterations of the forward–backward training, the parameters of the 48 phone models are trained.

These context-dependent phone models could be directly used in recognition or they could be used to initialize the training of the context-dependent phone models. In the second stage of context-dependent training, the same forward–backward algorithm is used. The only difference is that there are more context-dependent models, and that a context-dependent dictionary is used. For example, a context-independent dictionary expands *ship* into /sh ih p/, while a context-dependent dictionary might expand it into /sh(5) ih(9) p(2)/ where /sh(5)/ is one of many units for /sh/. The context-dependent models are trained with two more iterations of the forward–backward algorithm.

Since there is usually a large number of context-dependent models, many trained probabilities may be zeros. In order to estimate the probabilities of the unobserved and rare symbols, we interpolate the context-dependent model parameters with the corresponding context-independent ones. We use *deleted interpolation* [8] to derive appropriate weights in the interpolation. Deleted interpolation weighs each distribution according to its ability to predict unseen data. These weights can be estimated using the

TABLE II
List of the Set of Phones Used 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*

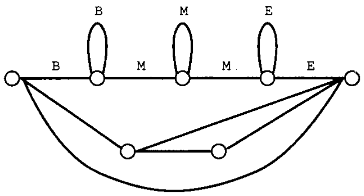

Fig. 2. Phone model used in SPHINX. Upper transitions are labeled by the output distribution to which they are tied. *B*, *M*, *E* correspond to the beginning, middle, and end of a phone. Lower transitions are assigned different labels depending on the phone.

forward-backward by viewing the weights as transition probabilities. (See [8] and [17] for more details.)

The SPHINX training procedure is shown in Fig. 3.

D. Recognition

Our recognition search is a time-synchronous Viterbi beam search [16] for the optimal state sequence. A beam threshold is determined *a priori*, and at a particular time, all states worse than the best state by more than that threshold are pruned. We also enhanced the Viterbi beam search with a word duration model when no grammar is used. The word duration model provides a word duration probability when the word is exited in the Viterbi beam search. This duration probability is combined with the acoustic probability.

Currently, SPHINX can recognize speech with the following language models.

- No language model—each word HMM (concatenated from phone HMM's) can transition into all other words. It has a perplexity of 997.
- A word pair language model—a word can only transition to words that can legally follow it; every pair of adjacent words must be legal in the grammar. All successors are considered equally likely. This language model has a perplexity of 60.
- A bigram language model—similar to word pair, except a probability for each successor is estimated from the underlying finite state grammar. The bigram language model has a perplexity of 20.

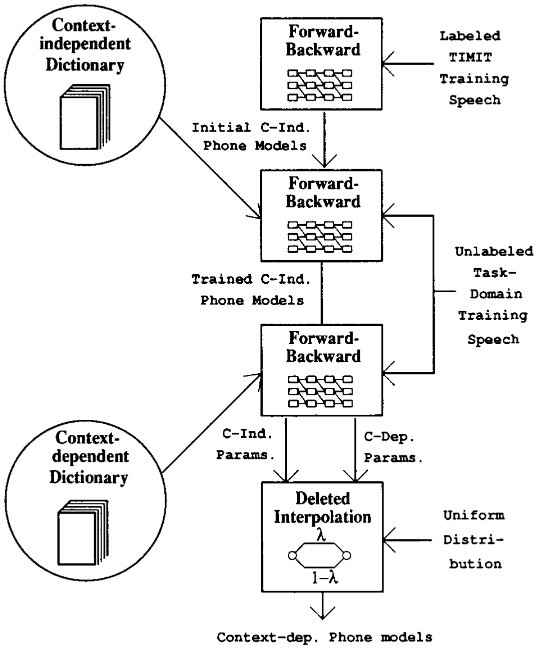
Context-independent Dictionary
Forward-Backward
Labeled TIMIT Training Speech
Initial C-Ind. Phone Models
Forward-Backward
Unlabeled Task-Domain Training Speech
Trained C-Ind. Phone Models
Forward-Backward
Context-dependent Dictionary
C-Ind. Params.
C-Dep. Params.
Deleted Interpolation
λ
1-λ
Uniform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

Fig. 3. The SPHINX training procedure.

IV. Results of Previously Proposed Subword Units

We applied various versions of SPHINX to the 997-word resource management task designated by DARPA [9]. All three grammars described in the previous sections were used. 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 [29]. This alignment determines *WordsCorrect, Substitutions, Deletions, Insertions*. Finally, *WordAccuracy* is computed by

$$\textit{WordAccuracy} = 100 \cdot \frac{\textit{CorrectLength-Subs-Dels-Ins}}{\textit{CorrectLength}} \quad (1)$$

where *CorrectLength* is the number of words in the correct sentence. Confusions between homonyms (such as *ship's* and *ships* or *two* and *too*) are not counted for the no language model, and are counted for the word pair and the bigram language model.

We used the 1987 DARPA speaker-independent test data for evaluation. This test data consists of 150 sentences, with 10 sentences spoken by each of 15 test speakers, who are not part of the training set. The same test set is used for all the results described in this paper.

The results with various context-dependent phonetic units are shown in Table III. Left-context modeling of a phone considers two phones with the same identity to be different if their left contexts are different. Left-context-dependent phone parameters are interpolated with context-independent phone parameters and a uniform distri-

TABLE III
WORD ACCURACY RESULTS WITH CONTEXT-INDEPENDENT AND CONTEXT-DEPENDENT PHONE MODELING

Version	Models	No grammar	Word pair	Bigram
Context-independent	48	49.6%	84.4%	90.6%
Left-context	787	61.6%	89.0%	93.8%
Right-context	786	62.1%	89.3%	94.0%
Triphone-context	2381	69.9%	92.2%	95.1%

TABLE IV
50 DIFFERENT WAYS *THE* WAS PRONOUNCED, ACCORDING TO SPECTROGRAM READERS

/dh ax/	/dh ix/	/- dh ax/	/- dh ix/	/dh iy/
/ax/	/- dh iy/	/dh ah/	/ix/	/th ax/
/dh ih/	/- d ix/	/iy/	/th ix/	/- dh ah/
/th iy/	/- d ax/	/- d ih/	/- d iy/	/- dh ih/
/dh/	/- dh eh/	/d ix/	/dh ao/	/dh uh/
/dx ax/	/ih/	/- d ah/	/- dh ao/	/- dh uh/
/- t ah/	/- t ih/	/- th iy/	/ah/	/d ih/
/d iy/	/dh ax q/	/dh er/	/dh iy ih/	/dh iy n/
/dh m/	/dx ah/	/dx ih/	/dx ix/	/eh/
/nx ah/	/nx ey/	/nx ix/	/th eh/	/ux/

bution. Right-context modeling is the same, except right contexts are used. The recognition rate improved substantially with the addition of left or right context, at the expense of 16 times as many models. Triphone context considers two phones with the same identity to be different when either the left or the right context is different. Similar to BBN's approach [1], triphone models are interpolated with left-context phone models, right-context phone models, context-dependent models, and a uniform distribution. Triphone models led to significantly better results than left or right context models, which illustrates the importance of *both* left and right contexts.

One difference between our approach and BBN's is that we use deleted interpolation to train the interpolation weights. Similar to other studies on triphones [1], [7], we have only considered intraword triphones, and a special word-boundary marker is used as the context for the boundary phones. Interword triphones will be explored in the future.

V. FUNCTION-WORD-DEPENDENT PHONE MODELS

Function words, such as *the*, *a*, *in*, *with*, are typically articles, prepositions, conjunctions, pronouns, and short verbs. These words are particularly problematic in continuous speech recognition. Waibel [30] found that in continuous speech, only 14% of the function words are stressed, while 93% of the content words are stressed. These unstressed syllables are much harder to recognize [31], [32]. While function words are spoken clearly in isolated-word speech, they are articulated extremely poorly in continuous speech. The phones in function words are distorted in many ways. They may be shortened, omitted, or seriously affected by neighboring contexts. For example, Table IV enumerates 50 phonetic transcription labels assigned by expert spectrogram readers for the word *the*. Many other function words have a large number of pronunciations. 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. Among the 684 errors in our system when no grammar was used, 334 were function word errors. Function words take up only 4% of the vocabulary or about 30% if weighted by frequency, yet they are accountable for almost 50% 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 used only for function words. This improves the modeling of the most difficult subset of words. Unlike word-dependent phones, function-word-dependent phones are easily trainable because function words occur frequently in any task. Finally, function-word-dependent phones can absorb multiple pronunciations, which are not explicitly modeled in SPHINX.

We selected a set of 42 function words 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 are appropriate for this task. These function words are shown in Table V.

Although function words are frequent, some distributions may still be undertrained, and some probabilities may have very small values. As with other context-dependent units, we use deleted interpolation to combine the function-word-dependent parameters with context-independent ones. Table VI shows the phones in *are* and *be*, the counts for the distributions of each phone, and the λ's for the function-word-dependent model parameters, phone model parameters, and uniform distribution. It can be seen that distributions (such as the last distribution of /b/ and the first distribution of /iy/ in *be*) are much more dependent on word context than their counts would have indicated.

Next, function-word-dependent phone models were used in conjunction with each of the context-dependent phone models. The results with and without function-word-dependent phone modeling are shown in Table VII. In each of the four cases, modeling function-word-dependent phones led to improvements. The improvement was the smallest for triphone contexts because about half of the phones in function words have unique triphone contexts, which means triphone modeling was already doing function-word-dependent phone modeling for some of the function words. We expect that function-word-dependent phone models will significantly improve systems that do not have very detailed phone models or systems with very large vocabularies where phones in function words would not be uniquely specified by triphones.

Table VIII 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%, which accounts for almost all of the improvement from 49.6 and 57.0% accuracy.

TABLE V
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

TABLE VI
λ'S TRAINED FOR PHONES IN FUNCTION WORDS *BE* AND *ARE*. λ_{wdep} IS THE WEIGHT FOR THE FUNCTION-WORD-DEPENDENT DISTRIBUTION, λ_{indep} IS THE WEIGHT FOR THE CONTEXT-INDEPENDENT DISTRIBUTION, AND $\lambda_{\mathrm{uniform}}$ IS THE WEIGHT FOR UNIFORM DISTRIBUTION

Word	Phone	Dist.	Count	λ_{wdep}	λ_{indep}	$\lambda_{uniform}$
ARE	/aa/	Begin	2333	0.788	0.173	0.039
		Middle	2025	0.706	0.284	0.010
		End	1266	0.830	0.127	0.043
	/r/	Begin	1513	0.890	0.084	0.026
		Middle	1794	0.904	0.092	0.004
		End	2016	0.814	0.173	0.013
BE	/b/	Begin	176	0.207	0.786	0.007
		Middle	249	0.263	0.732	0.005
		End	243	0.705	0.295	0.000
	/iy/	Begin	222	0.636	0.358	0.006
		Middle	571	0.348	0.651	0.000
		End	329	0.337	0.659	0.004

TABLE VII
IMPROVEMENT FROM FUNCTION-WORD-DEPENDENT PHONE MODELING. RESULTS SHOWN ARE WORD ACCURACY

Version	Models	No grammar	Word pair	Bigram
Context-ind.	48	49.6%	84.4%	90.6%
CI+fnwd-dep.	153	57.0%	87.9%	93.0%
Left-context	787	61.6%	89.0%	93.8%
LC+fnwd-dep.	892	66.6%	91.1%	94.7%
Right-context	786	62.1%	89.3%	94.0%
RC+fnwd-dep.	891	67.2%	91.5%	94.7%
Triphone-context	2381	69.9%	92.2%	95.1%
TC+fnwd-dep	2447	69.9%	92.4%	95.2%

TABLE VIII
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

VI. GENERALIZED TRIPHONES

In our evaluation of triphones, we argued that some phones have the same effect on neighboring phones. For example, the place of articulation has an important effect on the neighboring vowels. /b/ and /f/ have similar effects on the right-neighboring vowel, while /r/ and /w/ have similar effects on the right-neighboring vowel. Fig. 4 illustrates this phenomenon. If we could identify these similar contexts and merge them, we would have a much more manageable number of models, as well as much more training for each model.

One approach is to merge perceptually similar contexts together using human knowledge [33], [34]. This guarantees that the merged contexts are sensible ones; however, if we were to consider all triphones, this would be a complicated process. Moreover, while the example we gave is clear, there are many where there may be no consensus even among the experts. Therefore, we believe it is desirable to automate this process of context generalization.

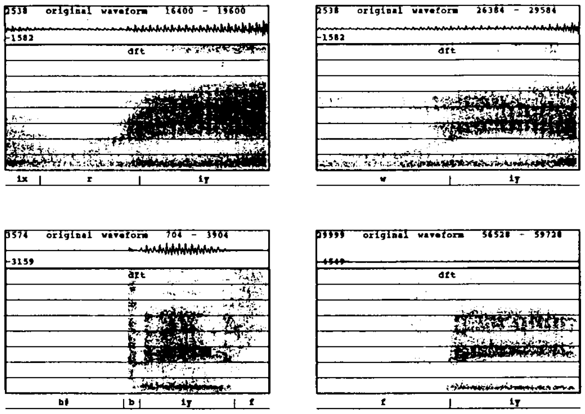

Fig. 4. The waveforms and spectrograms for the phoneme /iy/ with four different *left contexts* are illustrated. Note that /r/ and /w/ have similar effects on /iy/, while /b/ and /f/ have similar effects on /iy/. This illustrates that different left contexts may have similar effects on a phone.

We propose a context merging procedure to find and combine similar contexts.

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.

a) Move the element if the resulting configuration is an improvement.

b) Repeat until no such moves are left.

5) Until some convergence criterion is met, go to step 2).

Without step 4), this is simply an agglomerate clustering procedure [35] where every merge cannot be undone. Step 4) is a heuristic optimization that attempts to improve the clustering by allowing elements to be moved from one cluster to another. Although it appears expensive, if we remember which clusters have changed and which cluster pairs need not be compared, step 4) only triples the total computation for our task.

Many criteria could be used to determine the similarity between two HMM's. Juang and Rabiner [36] proposed several similarity measures using cross entropy, divergence, and discrimination information. Paul and Martin [7] investigated merging of continuous HMM's using a chi-square measure. Finally, D'Orta *et al.* [37] proposed several measures, including output string/symbol probability and maximum mutual information.

In this study, we use an information-theoretic measure that determines the similarity between two HMM's based on the amount of information lost when the two models are merged. We use entropy of the original and merged HMM's to measure the information lost, and forward-backward counts to weigh the information lost. Entropy clustering has been used by Lucassen and Mercer [38] to derive word baseforms from spelling, and by Brown [28] to merge similar VQ codewords.

In order to minimize the amount of computation, we define the entropy of an HMM as the bits of information in the output pdf's. Let

$N_{a,d}(i)$ be the count for codeword i in distribution d of context a of a phone as determined by the forward-backward algorithm

$$N_{a,d} = \sum_i N_{a,d}(i). \tag{2}$$

These counts can be normalized into output probabilities:

$$P_{a,d}(i) = \frac{N_{a,d}(i)}{N_{a,d}}. \tag{3}$$

The entropy of an output pdf for distribution d for some phone in context a defined as

$$H_{a,d} = -\sum_i P_{a,d}(i) \cdot \log\left(P_{a,d}(i)\right). \tag{4}$$

If we want to merge distribution d of two models that represent the same phone in context a and b into a merged model in context m, the new counts for distribution d in context m are simply

$$N_{m,d}(i) = N_{a,d}(i) + N_{b,d}(i). \tag{5}$$

We can compute $H_{b,d}$ and $H_{m,d}$ as we computed $H_{a,d}$. The information lost when a distribution d for context a and b are merged into m, weighted by counts, is

$$L_d(a, b) = N_{m,d}H_{m,d} - N_{a,d}H_{a,d} - N_{b,d}H_{b,d}. \tag{6}$$

Finally, the information lost when two HMM's for context a and b are merged, weighted by counts, is

$$L(a, b) = \sum_{d'} L_{d'}(a, b). \tag{7}$$

Equation (7) is the distance metric used in our triphone clustering algorithm. This distance metric weighs the difference between models according to the frequency of the models. By preferring to merge models that do not appear frequently, each generalized model will be more trainable.

For an alternate view of the distance metric, we transform (6) into

$$\begin{aligned}
L_d(a, b) &= N_{m,d}H_{m,d} - N_{a,d}H_{a,d} - N_{b,d}H_{b,d}\\
&= -N_{m,d} \sum_i P_{m,d}(i) \cdot \log P_{m,d}(i)\\
&\quad + N_{a,d} \sum_i P_{a,d}(i) \cdot \log P_{a,d}(i)\\
&\quad + N_{b,d} \sum_i P_{b,d}(i) \cdot \log P_{b,d}(i)\\
&= -\sum_i N_{m,d}(i) \cdot \log P_{m,d}(i) + \sum_i N_{a,d}(i)\\
&\quad \cdot \log P_{a,d}(i) + \sum_i N_{b,d}(i) \cdot \log P_{b,d}(i)\\
&= -\log\left(\prod_i \left(P_{m,d}(i)\right)^{N_{m,d}(i)}\right)\\
&\quad + \log\left(\prod_i \left(P_{a,d}(i)\right)^{N_{a,d}(i)}\right)\\
&\quad + \log\left(\prod_i \left(P_{b,d}(i)\right)^{N_{b,d}(i)}\right)\\
&= \log\\
&\left(\frac{\left(\overset{i}{\prod_i} \left(P_{a,d}(i)\right)^{N_{a,d}(i)}\right) \cdot \left(\prod_i \left(P_{b,d}(i)\right)^{N_{b,d}(i)}\right)}{\prod_i \left(P_{m,d}(i)\right)^{N_{m,d}(i)}}\right).
\end{aligned} \tag{8}$$

Equation (8) shows that the "weighted loss of information" measure is equivalent to the logarithm of the ratio between the probability that the individual distributions generated the training data and the probability that the combined distribution generated the training data. Thus, this ratio maximizes the objective function:

$$\prod_{\forall d} \prod_i \left(P_{m,d}(i)\right)^{N_{m,d}(i)} \tag{9}$$

which is consistent with the maximum likelihood criterion used in the forward-backward algorithm.

This context generalization algorithm provides a means for finding the equilibrium between trainability and specificity. Armed with this technique, we could empirically find the "right" number of models for any given amount of training data.

Table IX shows the 19 clusters created for the phone /ae/ when the clustering process has reduced 2381 triphones to 500 triphone clusters. Most clusters consist of triphones that are easily identified as similar contexts.

Results for generalized triphone modeling are shown in Table X. We ran the information-theoretic agglomerate clustering algorithm from 2381 triphones, and saved the clusters for every 100 merges. We then trained and tested

TABLE IX
19 Clusters Created for the (Left, Right) Contexts for Phone / ae /; # Represents Word-Boundary Context

[1]	(dh,td)
[2]	(hh,v) (hh,td)
[3]	(hh,z) (p,s)
[4]	(k,l) (g,l) (#,p)
[5]	(k,sh) (k,dx)
[6]	(l,n) (l,m)
[7]	(l,ch) (l,sh) (l,ts) (l,td) (l,dx) (l,s)
[8]	(m,k) (m,kd) (s,k)
[9]	(p,kd) (p,k) (t,k)
[10]	(k,n) (k,m) (k,r) (z,m) (ch,n) (s,m)
[11]	(r,th) (r,s) (r,f) (r,kd) (r,k)
[12]	(#,dx) (#,dd) (hh,dd) (b,jh) (b,dd)
[13]	(#,l) (#,f) (#,b) (b,sh) (m,dx) (n,sh) (hh,f)
[14]	(#,ae) (ay,m) (jh,n) (iy,kd) (y,ng) (d,g) (jh,k) (r,dd)
[15]	(#,td) (#,s)
[16]	(#,v) (#,r)
[17]	(v,l)
[18]	(s,dx) (d,dx) (f,s) (b,s) (k,td) (ch,dx) (s,s) (t,s) (k,z) (sh,s) (g,s) (k,s) (k,t) (k,th)
[19]	(z,n) (s,n) (v,n) (#,ng) (b,ng) (m,n) (f,n) (p,n) (r,n) (r,m) (er,m)

TABLE X
Results of Generalized Triphone Modeling without Function Word Modeling. Results Shown are Word Accuracy

Number of gen. models	No grammar	Word pair	Bigram
48	49.6%	84.4%	90.6%
100	60.9%	89.1%	94.0%
200	66.4%	91.0%	94.2%
300	66.2%	91.1%	94.1%
400	67.9%	91.8%	94.3%
500	69.6%	92.0%	95.1%
600	70.0%	92.4%	95.1%
800	70.3%	92.9%	95.1%
1000	70.3%	93.3%	95.4%
1200	70.0%	93.0%	95.3%
1400	69.7%	92.7%	95.2%
2381	69.9%	92.2%	95.1%

on ten different sets of generalized triphones, with 100, 200, 300, 400, 500, 600, 800, 1000, 1200, and 1400 models. We also included results with 48 HMM's (phone models, complete generation) and 2381 HMM's (triphone models, no generalization). In each case, the performance is superior to earlier context-dependent results that employed a comparable number of models. For example, with only 100–200 models, generalized triphones performed as well as 700–800 left or right context-dependent phones. Generalized triphone models also outperformed full triphone models, in spite of the fact that full triphone models were interpolated with three other types of models, while generalized triphone models were only interpolated with one. These results demonstrate the importance of modeling the left *and* right contexts. They also demonstrate that generalized triphones can improve the accuracy, while substantially reducing the memory requirements.

Table X illustrates that the performance of the system improved with more models until there were too many models to train adequately. At 1000 models, an equilibrium between trainability and specificity was reached, given the amount of training. We believe the performance can be improved with more models and more training.

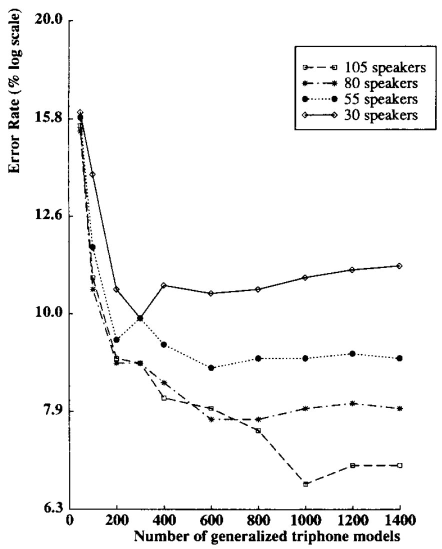

Fig. 5. Results with the word-pair grammar as a function of the number of generalized triphone models and the number of training speakers.

We performed an experiment where we reduced the number of speakers used in the training, and observed the effect of that on the optimal number of generalized triphones. Fig. 5 illustrates that as we reduced the amount of training, the best number of generalized triphones decreased because fewer models can be adequately trained. This further justified the use of generalization to find an equilibrium between trainability and specificity.

Finally, we added function-word-dependent phone modeling to triphone clustering, and ran an experiment with 1000 generalized triphone models plus 153 function-word-dependent phone models. Since some of the function-word-dependent phones are uniquely specified by the generalized triphones, there are a total of 1076 models. The results with these 1076 models are shown in Table XI.

VII. Future Work

There are many interesting areas of continuing research in context-dependent phone modeling. We believe that these research areas will improve our results considerably.

The triphone-based models used in SPHINX and other systems stop at word boundaries. In other words, the leftmost phone of each word has no known left context, and the rightmost phone of each word has no known right context. Yet, phonetic context beyond the word boundary clearly affects the realization of these word-boundary triphones. This suggests that modeling interword triphones should lead to superior results. We have recently implemented between-word triphone models, and preliminary results indicate that the error rate can be further reduced

TABLE XI
RESULTS WITH GENERALIZED TRIPHONE MODELING PLUS FUNCTION WORD MODELING. RESULTS SHOWN ARE WORD ACCURACY

Version	Models	No grammar	Word pair	Bigram
Gen. Triphones	1000	70.3%	93.3%	95.4%
Gen. Triphones +fnwd-dep.	1076	70.6%	93.7%	95.8%

by about 20%. Details of this work will appear in [39], [40].

Another research area is the extension from *generalized triphones* to *generalized allophones*. The use of generalization from specific units need not be limited to triphones. As we obtain more training data, triphone models will be trainable without the need of generalization. However, at that time, we believe it will be unwise to be complacent with well-trained triphone models. Instead, we should consider additional sources for phonetic variability, such as syllable position, stress, nonneighboring phones, or interword triphones [41]. The context generalization technique we described could be used to merge these more detailed allophones into generalized allophones.

Finally, we would like to investigate the vocabulary dependence of context-dependent models—in other words, how well context-dependent models trained from one vocabulary will work on another vocabulary. Our initial results indicate serious performance degradation from training on a considerably different vocabulary. We believe that this is due to inadequate test set triphone coverage in the training set. Our approach to dealing with this problem is to train context-dependent models from a large database of "general English," and then apply them to specific tasks. We are currently collecting such a database for this experiment. We hope that the availability of a large database and more detailed generalized allophone models can compensate for the lack of vocabulary-specific training. If successful, this would enable speaker-independent speech recognition *without vocabulary-specific training.*

VIII. CONCLUSION

Speech production is a complex process where the acoustic realizations of different parts of a sentence are correlated and interdependent. Thus, the only way to ensure preservation of all the information is to model every possible sentence. Since there is an astronomical number of sentences, it is necessary to use concatenated words as sentences. Similarly, the acoustic realizations of different parts of a word are highly interdependent, and there are still too many words to train adequately with the current technology and databases. Therefore, it is necessary to use concatenated subword units as words.

Because most interdependencies are local, using larger units would capture most of the important effects; however, as the units grow in size, they also grow in number, which causes trainability problems. Using smaller units, on the other hand, sacrifices important information by combining many different effects into one representation.

Context-dependent phonetic units are a compromise between specificity and trainability. By modeling context-dependent effects at the phone level, these units achieve the specificity needed to make fine phonetic distinctions. Yet, because they are phonetic units, they can be interpolated with context-independent phones for trainability. The most popular forms of context-dependent phone modeling have been triphones, which take into consideration the immediate left and right contexts of a phone.

In this work, we have shown the feasibility of context-dependent phone modeling to speaker-*independent* recognition. We have also proposed two new context-dependent units. The first unit is the function-word-dependent phone, which explicitly models phonetic events in the poorly articulated, but frequent function words. This improves the discriminability of these function words. The other unit is the generalized triphone, which combines similar triphones together to improve trainability of the models. An information-theoretic distance metric was used to cluster the models. This metric was shown to be consistent with the maximum likelihood criterion. Thus, this technique enables us to find a set of models that is as consistent and trainable as possible, given a fixed amount of training data.

Both function-word-dependent phone modeling and generalized triphone modeling improved recognition accuracy substantially from context-independent phone modeling. In particular, function-word-dependent phone modeling recovered a large number of function word errors. Generalized triphone models also led to higher recognition accuracies than triphone models, while saving 60% memory. We believe context-dependent phonetic modeling have potential for further improvements. Our future work will involve extending context-dependent phonetic models for between-word coarticulation modeling, generalized allophone modeling, and vocabulary-independent modeling.

ACKNOWLEDGMENT

The author wishes to thank H.-W. Hon, M.-Y. Hwang, R. Reddy, P. Brown, and R. Schwartz for their help and discussions.

REFERENCES

[1] 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.

[2] R. P. Lippmann, E. A. Martin, and D. P. Paul, "Multi-style training for robust isolated-word speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1987, pp. 705–708.

[3] 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.

[4] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Further results on the recognition of a continuously read natural corpus," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1980.

[5] R. M. Schwartz, Y. L. Chow, S. Roucos, M. Krasner, and J. Makhoul, "Improved hidden Markov modeling phonemes for continuous speech recognition," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1984.

[6] K. F. Lee and H. W. Hon, "Large-vocabulary speaker-independent continuous speech recognition," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1988.

[7] 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.

[8] 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, Ed. Amsterdam: North-Holland, 1980, pp. 381-397.

[9] 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.

[10] 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.

[11] M. J. Hunt, M. Lennig, and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1980, pp. 880-883.

[12] 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.

[13] R. Schwartz, J. Klovstad, J. Makhoul, and J. Sorensen, "A preliminary design of a phonetic vocoder based on a diphone model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1980, pp. 32-35.

[14] D. Klatt, "Problem of variability in speech recognition and in models of speech perception," in *Variability and Invariance in Speech Processes*, J. S. Perkell and D. M. Klatt, Ed. Hillsdale, NJ: Lawrence Erlbaum Assoc., 1986, pp. 300-320.

[15] M. Cravero, R. Pieraccini, and F. Raineri, "Definition and evaluation of phonetic units for speech recognition by hidden Markov models," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 1986.

[16] 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.

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

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

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

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

[21] 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.

[22] K. F. Lee and H. W. Hon, "Speaker-independent phoneme recognition using hidden Markov models," Dep. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-88-121, Apr. 1988.

[23] 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.

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

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

[26] 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.

[27] 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.

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

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

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

[31] D. H. Klatt and K. N. Stevens, "Sentence recognition from visual examination of spectrograms and machine-aided lexical searching," in *Proc. 1972 Conf. Speech Commun. Processing*, IEEE and AFCRL, 1972, pp. 315-318.

[32] W. A. Lea, *Trends in Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[33] A.-M. Derouault, "Context-dependent phonetic Markov models for large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1987, pp. 360-363.

[34] L. Deng, M. Lennig, V. N. Gupta, and P. Mermelstein, "Modeling acoustic-phonetic detail in an HMM-based large vocabulary speech recognizer," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1988, pp. 509-512.

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

[36] B. H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden Markov models," *Bell Syst. Tech. J.*, vol. 64, pp. 391-408, Feb. 1985.

[37] P. D'Orta, M. Ferretti, and S. Scarci, "Phoneme classification for real time speech recognition of Italian," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1987, pp. 81-84.

[38] J. M. Lucassen and R. L. Mercer, "An information theoretic approach to the automatic determination of phonemic baseforms," presented at the IEEE Int. Conf. Acoust., Speech, Signal Processing, 1984.

[39] M. Y. Hwang, H. W. Hon, and K. F. Lee, "Between-word coarticulation modeling for continuous speech recognition," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep., Mar. 1989.

[40] ——, "Modeling between-word coarticulation in continuous speech recognition," in *Proc. Eurospeech*, Sept. 1989.

[41] F. Chen and P. E. Stern, "Contextual variability in speech classification," presented at the IEEE Workshop on Speech Recognition, June 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.