HIDDEN MARKOV MODELS:
PAST, PRESENT, AND FUTURE

Kai-Fu Lee
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract
Hidden Markov models (HMMs) have recently become the predominant approach to speech recognition. In this paper, we will elucidate the success of hidden Markov models, using Sphinx as an example of an HMM-based system. We will also outline avenues for future research in hidden Markov models.

1. Introduction
Hidden Markov models (HMMs) have recently become the predominant approach to automatic speech recognition. HMM-based systems make certain structural assumptions, and then try to learn two sets of parameters from training data. The forward-backward learning algorithm adjusts these model parameters so as to improve the probability that the models generated the training data. This seemingly simple technique has worked surprisingly well, and has led to many state-of-the-art systems [1, 2, 3, 4, 5, 6] that far outperformed other techniques.

Some of the questions often raised about HMMs include:
- Why have HMMs worked so well?
- What are the key factors in building a successful HMM-based speech recognition system?
- Have HMMs been pushed to the limit, or is there still room for further improvement?

This paper will attempt to answer the first two questions, and shed light on the third.

In Section 2, we will briefly introduce hidden Markov modeling, and explain why they are a powerful modeling technique for time-varying signals such as speech.

In Section 3, we will describe the three key factors for a successful HMM system: plentiful training data, a powerful learning algorithm, and detailed modeled. We will use Sphinx [7, 5], our large-vocabulary speaker-independent continuous speech recognizer, as an example to illustrate the contributions of each factor.

Finally, we believe that HMMs have not yet realized their full potential, and there are still many unexplored areas that could further advance the state of the art. In Section 4, we will identify some of these areas that we are currently exploring at Carnegie Mellon.

2. Hidden Markov Models

2.1. A Brief Introduction to HMMs
Hidden Markov models (HMM) were first described in the classic paper by Baum [8]. Shortly afterwards, they were extended to automatic speech recognition independently at CMU [9] and IBM [10, 11]. It was only in the past few years, however, that HMMs became the predominant approach to speech recognition, superseding dynamic time warping.

A hidden Markov model is a collection of states connected by transitions. Each transition carries two sets of probabilities: a transition probability, which provides the probability of taking this transition, and an output probability density function (pdf), which defines the conditional probability of emitting each output symbol from a finite alphabet, given that that the transition is taken. Figure 1 shows an example of a hidden Markov model with two output symbols, A and B.

There are several types of hidden Markov models. The simplest and most natural one is discrete density HMMs, which are defined 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 \( j \).
- \( \{b_j(k)\} \) — The output probability matrix: the probability of emitting symbol \( k \) when taking a transition from state \( i \) to state \( j \).

The forward-backward algorithm is used to estimate \( a \) and \( b \). We provide only a simplistic sketch here; details of the algorithm can be found in [12, 7]. The forward-backward algorithm adjusts \( a \) and \( b \) iteratively. For each iteration, the
estimates from the previous iteration are used to count how frequently each symbol is observed for each transition, and how frequently each transition is taken from each state. These counts are then normalized into new parameters. Let $c_{ij}(k)$ represent the frequency (or count) that the symbol $k$ is observed, and that the transition from $i$ to $j$ is taken, the new output probability $\bar{b}_{ij}(k)$ is given by the normalized frequency:

$$\bar{b}_{ij}(k) = \frac{c_{ij}(k)}{\sum_{k'=1}^{K} c_{ij}(k')} \quad (1)$$

Similarly, transition probabilities are re-estimated by normalizing the frequency that a transition is taken from a particular state:

$$\bar{\alpha}_{ij} = \frac{\sum_{k=1}^{K} c_{ij}(k)}{\sum_{i,j} \sum_{k=1}^{K} c_{ij}(k')} \quad (2)$$

Baum [8] showed that re-estimating $a$ and $b$, as shown in equations 1 and 2, will increase the likelihood of generating the training data, unless a local maximum has been reached. Although the forward-backward algorithm guarantees only a local maximum, it efficiently produces an approximation to the maximum-likelihood estimates (MLE) of the HMM parameters.

2.2. Advantages of HMMs for Speech Recognition

In the previous section, we have introduced the basic mechanism of hidden Markov models. It has been, and probably still is, surprising to many that such a simple modeling technique has led to the highest performance systems in almost every speech recognition problem today. In this section, we will try to explain why HMMs have worked so well.

Hidden Markov models have a rich representation in their two sets of parameters. The output probabilities represent the acoustic phenomena. They could be based on either discrete densities, where speech is quantized into sequences of symbols from a finite alphabet (usually through vector quantization). Alternatively, they could be based on continuous mixture densities (usually Gaussian), where speech parameters are directly modeled. In either case, they have the power of modeling any arbitrary probability density function, given sufficient training data. The other set of parameters, the transition probabilities, represent timescale distortions. With a large number of states, duration of very fine phonetic events can be modeled. Yet, with the use of self-loops, the range of durations modeled is very large. Finally, the joint optimization of the two sets of parameters makes HMMs particularly suitable for modeling of time-varying signals.

Speech recognition is a very difficult problem. There are many sources of variability (speaker, context, environment, accent, etc.), which can only be modeled with a large amount of training data. Since hand-tuning on a large database is impossible, it is desirable for a speech recognizer to be able to automatically generalize from a large amount of data. Hidden Markov models have such a capability by first making structural assumptions, and then using parameter estimation to improve the probability that the models generated the training data. This learning process, the forward-backward algorithm has many desirable properties:

- It requires minimal supervision — only an orthographic transcription of the speech is needed.
- It has a mathematical basis, guaranteeing convergence to a critical point.
- It scales gracefully to increased training, requiring only linearly more computation.

Finally, speech recognition requires reasoning with uncertainty, and since probabilities are the mathematics of uncertainty, the probabilistic nature of HMMs makes them the ideal representation for speech recognition.

Speech recognition involves a search in a state-space for an optimal, or a near-optimal solution. HMM-based searches differ from bottom-up approaches, which propagate errors and cannot integrate top-down knowledge, and from top-down approaches, which are often intractable. It is possible to represent sounds, phonemes, syllables, words, and even grammar states in terms of HMMs. By integrating many knowledge levels into a unifying HMM framework, the HMM search is a global, goal-driven strategy, where all knowledge sources participate in every decision. Finally, by using a probabilistic framework, we have a consistent scoring mechanism.

In summary, hidden Markov models have a number of very powerful properties. The ability of HMMs to automatically optimize parameters from data is extremely powerful, the HMM integrated search that considers all of the knowledge sources at every step is very effective, and the absorption of faulty structural assumptions is most forgiving. By turning an unknown structure problem into an unknown parameter problem, and by automatically optimizing these parameters, HMM and maximum likelihood estimation are one of the most powerful learning paradigms available today.

3. Recent HMM Improvements for Speech Recognition

In the previous section, we have explained why hidden Markov models are particularly suitable for modeling speech. The single greatest advantage of hidden Markov models is the existence of an automatic training algorithm. However, it does not imply that HMMs are completely self-organizing tools. In fact, literature is full of examples where simple-minded HMMs produce very poor results. For example, the first recognition rate we attained on the 991-word, perplexity 60 task was only 58% [5] (compared to the recent result of 96%). In this section, we will examine a number of enhancements to the HMM paradigm. We will focus on the enhancements and effects on the speaker-independent Sphinx continuous speech recognition system.
We believe that these enhancements can be categorized into:

- Detailed speech models.
- Large training databases.
- Improved learning algorithms.

In the next three sections, we will discuss how these factors contributed to Sphinx.

3.1. Detailed Speech Models

By "detailed speech models", we mean the expansion of the HMM parameters or modification of the HMM structures selectively. Intuitively, both should be helpful to HMMs. Expansion of the parameters should improve performance, assuming sufficient training data are available. Improving the HMM structures should also be helpful, since the HMM learning process assume the correctness of fixed underlying structures.

However, we have found that the amount of improvement greatly depends on careful selection of the right parameters to expand, and the right structures to tune. In particular, we have found two types of improvements to be the most helpful:

- Detailed models that compensate for the weaknesses of HMMs.
- Detailed models that utilize our speech knowledge.

In this section, we will examine how these enhancements improved the Sphinx Speech Recognition System.

Multiple Codebooks

Discrete hidden Markov models model speech as a sequence of vector quantized symbols. In other words, every frame of speech is reduced to a symbol from a finite alphabet. Typically a single codebook using stationary coefficients (FFT, LPC, etc.) is used. However, it has been shown recently that the use of differential information and power information is extremely important [13]. Moreover, one of the serious problems with HMMs is that they assume that speech events are only dependent on the state, which causes the HMMs to have no vision of the past or the future. The inclusion of differential coefficients give HMMs more scope than a small 20-msec frame.

One possible approach to incorporate the differential and power coefficients is to use a single monolithic codebook. However, in such a codebook with many dimensions, the VQ distortion is very large. Therefore, we used the multiple-codebook approach [14]. Multiple codebooks reduce the VQ distortion by reducing the dimensions of the parameter space. We created 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.

Because we use three VQ codebooks, our discrete HMM must produce three VQ symbols at each time frame. By assuming that the three output pdf's are independent, we can compute the output probability as the product of the three output probabilities. The use of multiple codebooks increased Sphinx's accuracy from 26% to 45% (no grammar), and from 58% to 84% (word pair grammar).

Duration Modeling

For recognition, we have used a Viterbi search [15] that finds the optimal state sequence in a large HMM network. At the highest level, this HMM is a network of word HMMs, arranged according to the grammar. Each word is instantiated with its phonetic pronunciation network, and each phone is instantiated with the corresponding phone model. Beam search [16, 17] is used to reduce the amount of computation.

One problem with HMMs is that they do not enforce any global durational constraints. For example, a 50-state word HMM may have reasonable state durations at all 50 states, but the word duration may be unreasonable. To add this higher-level constraint, 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. This duration model resulted in a substantial improvement when no grammar is used (45% to 50%), but not when a grammar is used.

Function Word and Phrase Modeling

One problem with continuous speech is the unclear articulation of function words, such as a, the, in, of, etc. Since the set of function words in English is limited and function words occur frequently, it is possible to model each phone in each function word separately. By explicitly modeling the most difficult sub-vocabulary, recognition rate can be increased substantially. We selected a set of 42 function words, which contained 105 phones. We modeled each of these phones separately.

We have found that function words are hardest to recognize when they occur in clusters, such as that are in the. The words are even less clearly articulated, and have strong inter-word coarticulatory effects. In view of this, we created a set of phone models specific to function phrases, which are phrases that consist of only function words. We identified 12 such phrases, modified the pronunciations of these phrases according to phonological rules, and modeled the phones in them separately. A few examples of these phrases are: is the, that are, and of the.

Modeling these frequently occurring words and phrases increased the number of parameters by a factor of five, and improved Sphinx's accuracies from 50% to 59% (no grammar), and from 84% to 88% (word pair grammar).

Generalized Triphones

The function-word and function-phrase dependent phone models provide better representations of the function words. However, simple phone models for the non-function words are inadequate, because the realization of a phone crucially
depends on context. In order to model the most prominent contextual effect, Schwartz, et al. [17] proposed the use of triphone models. A different triphone model is used for each left and right context. While triphone models are sensitive to neighboring phonetic contexts, and have led to good results, there are a very large number of them, which can only be sparsely trained. Moreover, they do not take into account the similarity of certain phones in their affect on other phones (such as /b/ and /p/ on vowels).

In view of this, we introduce the generalized triphone model. Generalized triphones are created from triphone models using an agglomerative clustering procedure that clustered triphone models together using the following distance metric:

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

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 count of codeword \(i\) in model \(a\). \(m\) is the merged model by adding \(N_a\) and \(N_b\). This equation 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. Thus, it is consistent with the maximum-likelihood criterion used in the forward-backward algorithm.

This context generalization algorithm enables us to empirically determine how many models could be trained given a training set. Generalized triphones further increased the number of parameters by a factor of five, and improved the results from 59% to 79% (no grammar), and 88% to 94% (word pair grammar). Details of the context-dependent models used in Sphinx can be found in [5, 18].

### Between-Word Coarticulation Modeling

Triphone and generalized triphone models are powerful subword modeling techniques because they account for the left and right phonetic contexts, which are the principal causes of phonetic variability. However, these phone-sized models consider only intra-word context. A simple extension of triphones to model between-word coarticulation is problematic because the number of triphone models grows sharply when between-word triphones are considered. For example, there are 2381 within-word triphones in our 997-word task. But there are 7057 triphones when between-word triphones are also considered.

Therefore, generalized triphones are particularly suitable for modeling between-word coarticulation. We first generated 7057 triphone models that accounted for both intra-word and inter-word triphones. These 7057 models were then clustered into 1100 generalized triphone models. Few program modifications were needed for training, since the between-word context is always known. However, during recognition, most words now have multiple initial and final states. Care must be taken to to ensure that each legal sentence has one and only one path in the search. Details of our implementation can be found in [19]. The use of between-word coarticulation did not increase the number of parameters, since we felt that we could not reliably train any more parameters using our training database. Yet, Sphinx’s accuracies were improved from 73% to 78% (no grammar), and 94% to 95.5% (word pair grammar).

### 3.2. Large Training Database

A large database of 4200 sentences from 105 speakers were used to train Sphinx. Although this database is crucial to the success of Sphinx, it is more important to derive a system configuration that has enough parameters to model the variabilities in the data, but not too many parameters that we cannot reliably estimate. For example, if we fix the system configuration and reduce the training data, results deteriorate much faster than if we use a configuration that is data-dependent (such as using fewer generalized triphones). This phenomenon is clearly demonstrated in Figure 2.

![Figure 2: Sphinx accuracies with different number of training speakers and parameters.](image-url)

Therefore, while HMM system benefit greatly from increased training, it is inadequate to simply increase the training data. Instead, data-dependent system configuration is needed to optimize the performance. We have described some of these techniques in the previous section, and we will outline our future work in Section 4.1 and 4.2.

### 3.3. Better Learning Algorithms

Hidden Markov models with maximum-likelihood estimation (MLE) constitute the predominant approach to automatic speech recognition today. Indeed, the forward-backward algorithm is responsible for the success of Sphinx and many other systems. However, one of the problems with MLE is that it may produce inferior results when the underlying
models are incorrect, which HMM's obviously are as models of real speech. Thus, alternate training algorithms that do not suffer from this problem may be desirable. We have only experimented with one variant — corrective training, which will be described below.

### Corrective Training

Bahl et al. [20] introduced the corrective training algorithm for HMMs as an alternative to the forward-backward algorithm. While the forward-backward algorithm attempts to increase the probability that the models generated the training data, corrective training attempts to maximize the recognition rate on the training data. This goal has a definite practical appeal, since error rate, not sentence likelihood, is the bottom line for speech recognition. This algorithm has two components: (1) error-correction learning — which improves correct words and suppresses misrecognized words, (2) reinforcement learning — which improves correct words and suppresses near-misses.

We extended this corrective training algorithm to speaker-independent continuous speech recognition. We used a large training database and a cross-validated training procedure that fully made use of the training material. More importantly, we proposed a reinforcement learning method that hypothesized near-miss sentences by first formulating a list of near-miss phrases using a DTW algorithm, and then creating the near-miss sentences by substituting phrases from the near-miss phrase list.

Using this training algorithm, we were able to further improve our results from 78% to 82% (no grammar), and 95.5% to 96.2% (word-pair grammar). More details about this work are described in the proceeding [21].

### 3.4. The Sphinx System

By discussing various methods of generating detailed HMMs, we have uncovered most of the Sphinx System. Sphinx is trained by first using a set of context-independent models. Next, the context-dependent (function word/phrase dependent, generalized triphone, and between-word triphone) models are trained. Then, the parameters are smoothed using deleted interpolation [22], and corrective training is applied to improve discrimination. This training procedure is shown in Figure 3.

The Sphinx System was trained on about 105 speakers and 4200 sentences. It was tested on 150 sentences from 15 speakers. These sentences were the official DARPA test data for evaluations in March and October 1987. The word accuracies for various versions of Sphinx with the word-pair grammar (perplexity 60) and the null grammar (perplexity 997) are shown in Table 1. Word accuracy is defined as the percent of words correct minus the percent of insertions.

### 4. Future Outlook of HMMs

Sphinx is one of the many successful speech recognition systems [1, 2, 3, 4, 5, 6] that emerged in the past few years. While these systems have achieved record performance in many applications, they are still substantially worse than humans. The natural question to ask is: will HMMs approach human performance, or have they been pushed to the limit?

We believe hidden Markov modeling is a powerful approach that has not yet realized its full potential. Our previous experience has indicated that HMMs benefited from: detailed speech models, large training databases, and powerful learning algorithms. We feel that many promising improvements lie ahead in all three areas.
4.1. Detailed Speech Models

In the foreseeable future, we expect to continue to use context-dependent phonetic models. Currently, context-dependency includes only left context, right context, and word boundary. In practice, there are many other causes of phonetic variability, which can be classified into three categories:

- Articulation-related variabilities are caused by the fact that our articulators cannot move instantaneously, and that the locations and movements of the articulators affect the realization of a phone.
- Language-related variabilities are caused by attributes of specific languages.
- Speaker-related variabilities result from differences in anatomical features.

These three sources of variability modify various attributes and qualities of each phone, and we call the resultant of this transformation an allophone. The factors affecting the process of transforming a phone to an allophone is illustrated in Figure 4.

![Figure 4: Sources of variability that affect the realization of a phone.](image)

To make speech units completely consistent, it is necessary to explicitly consider each source of the above sources of phonetic variability. However, the number of all such combinations is astronomical, and we must improve by selective modeling the more relevant sources.

First, we will use our speech knowledge to identify and model only the most relevant contexts. For example, in order to collect a large database for this training, we will not model speaker-related variabilities. We will model immediate phonetic context, word/syllable boundary, and stress. We will also selectively model other contextual effects, such as non-neighboring phonetic contexts only when they are relevant. This strategy allows us to reduce an astronomical number of models into a more reasonable number (about 50,000).

50,000 subword models are still two orders of magnitude more than our current system can learn. Since we do not expect to have two orders of magnitude more training, we must further reduce these models to a more manageable level. One possibility is to extend the notion of bottom-up subword clustering as used in generalized triphones [5, 18]. We call this set of phonetic models generalized allophones. The clustering procedure will combine allophone models in order to maximize the probability that they generated the training data. The precise number of generalized allophones is data-dependent, and can be determined empirically.

The bottom-up subword clustering process finds a good mapping for each of the 50,000 allophones. However, if a context is not covered by these allophones, the context-independent phone model must be used instead, which will lead to substantially degraded performance. In other words, bottom-up clustering does not facilitate generalization; therefore, its utility will be determined by the allophonic coverage in the training data.

Another approach that sacrifices some optimality to improve generalization is the use of decision trees [23, 24] to cluster subword models. At the root of the decision tree is the set of all allophones corresponding to a phone. The algorithm incrementally splits nodes in the tree by asking "questions". These questions might be general ones like "is the previous phone a front vowel", or specific ones like "is the next phone in the set /p, t, k/ or the set /b, d, g/?". These questions will lead to a set of leaf nodes, which represent the contextual units to be used. This type of top-down subword clustering has two important advantages. First, if a new allophone is encountered, we might still be able to reach a leaf node, if all questions are sufficiently general. Even if unanswerable question is encountered at an internal node, we can still use that node as a subword unit, which should be much more appropriate than backing off the the context-independent phone. Second, since a child node must be somewhat similar to its parent node, we can improve trainability by interpolating each node with all of its ancestor nodes. One disadvantage of the top-down approach is that it may lead to a lower likelihood.

We hope that by making subword models more consistent and detailed, we will not only improve performance, but also have models that are more vocabulary-independent. This issue is discussed in more detail in [25].

4.2. Large Training Database

It is well known that a fixed statistical learning system will improve with additional training, until all the parameters are well-trained. Thus, one might conclude that additional training will help a speech recognizer only until an asymptote is reached. However, this statement is only true for a recognizer with fixed structures and parameter size. In reality, we are free to improve the structures and to increase the number of parameters of a recognition system, and we have seen that in so doing, the recognizer performance can be improved substantially.

In the previous sections, we have presented some ideas on how to make phonetic models more trainable with more data.
However, we aimed our research given the amount of training that is likely to become available in the next few years, or about tens of thousands of sentences. In the future, with the use of computers that have voice capabilities, data collection will become much easier. In a decade from now, we expect to see several orders of magnitude more training data. These data can be utilized to further refine the speech models. Table 2 shows the types of models that might be trainable with these future speech databases.

<table>
<thead>
<tr>
<th>Number of Sentences</th>
<th>Type of Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 100</td>
<td>Phonetic models</td>
</tr>
<tr>
<td>&gt; 1,000</td>
<td>Phonetic models with simple contexts (e.g. triphones).</td>
</tr>
<tr>
<td>&gt; 10,000</td>
<td>Phone models with more contexts (e.g. stress, syllable/word position).</td>
</tr>
<tr>
<td>&gt; 100,000</td>
<td>Longer (e.g. syllable) models; Rough speaker cluster (e.g. gender) models.</td>
</tr>
<tr>
<td>&gt; 1,000,000</td>
<td>Even longer (e.g. morph, word) models; Detailed speaker cluster (e.g. dialect) models.</td>
</tr>
</tbody>
</table>

Table 2: Types of models that might be trainable as the number of training sentences is increased in a speaker-independent database.

A speaker-independent HMM system is ideal for a large database with millions of sentences. Since the database can only be collected incrementally, it must be speaker-independent. Moreover, such a database can only be utilized by automatic learning techniques that scale well, which is certainly the case with HMMs. Finally, since we cannot afford to label the phonemes or even the words, the automatic-alignment inherent in the forward-backward algorithm is ideal.

Thus, we believe larger databases are essential for HMM systems, and HMM systems ideal for larger databases. We have recently begun an effort to collect a general English database [25], which we hope will serve as a starting point of an ambitious data collection effort.

4.3. Better Learning Algorithms

In our future research, we would like to improve HMM learning in three directions: (1) a more integrated learning framework, (2) use of discriminant learning, and (3) speaker adaptive learning.

One of the main advantages of HMMs is the integrated learning approach, where output probabilities and transition probabilities for all units are learned to improve a global measure. However, if we examine systems like Sphinx, there are at least two areas that are detached, and learning is impossible. First, the vector quantization process is a preprocessing stage that uses a distance metric not related to the MLE criterion. In order to rectify this problem, we have begun to use semi-continuous HMMs [26] that enable the learning of the vector quantization. The second detached element is the pronunciation dictionary, which maps words to phones. This dictionary is created from phonetic knowledge alone. The integration of dictionary learning with HMM learning should lead to further improvements.

The second area of research involves the incorporation of discrimination in the HMMs. To that end, we have used the corrective training algorithm [21]. Many other promising techniques, such as maximum mutual information estimation [27], and linear discriminants [6] have been introduced. Since we will not work on the "correction" of HMM assumptions in order to make MLE more valid, it is important to consider alternative approaches that do not depend on the problematic HMM assumptions. We are beginning to investigate the incorporation of above techniques into Sphinx. We are also investigating the possibility of integrating HMMs with neural networks. A preliminary study of that has been reported in [28].

Finally, we have only addressed the issue of HMM learning when presented with a large amount of multi-speaker training data for a one-time training process. In reality, few applications require true speaker-independence. There are usually opportunities to adapt on a small number of utterances from each speaker. Our previous work on speaker adaptation [5] have only led to modest error reductions (about 10%) with substantial adaptation (30 sentences). In the future, we must explore alternative approaches that can incrementally adapt more accurately on less data.

5. Conclusion

In this paper, we have presented the hidden Markov model methodology, using Sphinx as an example of a state-of-the-art HMM recognition system. We discussed why HMMs have worked so well, and outlined areas of future research. We believe that hidden Markov models have benefited greatly from the use of detailed subword models, large training databases, and powerful learning techniques. Further, we believe that HMMs have not yet realized their full potential, and that by expanding in each of the three areas, more advances are yet to come.

Acknowledgments

The author wishes to thank the CMU Speech Group for their support and contributions. This research was sponsored by Defense Advanced Research Projects Agency Contract N00039-85-C-0163.

References


*Although we map the phones to generalized triphones using a consistent distance metric, the original phone sequences are unrelated to the global optimization.


