*Proc. Natl. Acad. Sci. USA*
Vol. 92, pp. 9956–9963, October 1995
Colloquium Paper

***This paper is a condensed version of one that was presented at a colloquium entitled "Human–Machine Communication by Voice," organized by Lawrence R. Rabiner, held by the National Academy of Sciences at The Arnold and Mabel Beckman Center in Irvine, CA, February 8–9, 1993.***

# State of the art in continuous speech recognition

JOHN MAKHOUL AND RICHARD SCHWARTZ

BBN Systems and Technologies, Cambridge, MA 02138

**ABSTRACT** **In the past decade, tremendous advances in the state of the art of automatic speech recognition by machine have taken place. A reduction in the word error rate by more than a factor of 5 and an increase in recognition speeds by several orders of magnitude (brought about by a combination of faster recognition search algorithms and more powerful computers), have combined to make high-accuracy, speaker-independent, continuous speech recognition for large vocabularies possible in real time, on off-the-shelf workstations, without the aid of special hardware. These advances promise to make speech recognition technology readily available to the general public. This paper focuses on the speech recognition advances made through better speech modeling techniques, chiefly through more accurate mathematical modeling of speech sounds.**

More and more, speech recognition technology is making its way from the laboratory to real-world applications. Recently, a qualitative change in the state of the art has emerged that promises to bring speech recognition capabilities within the reach of anyone with access to a workstation. High-accuracy, real-time, speaker-independent, continuous speech recognition for medium-sized vocabularies (a few thousand words) is now possible in software on off-the-shelf workstations. Users will be able to tailor recognition capabilities to their own applications. Such software-based, real-time solutions usher in a whole new era in the development and utility of speech recognition technology.

As is often the case in technology, a paradigm shift occurs when several developments converge to make a new capability possible. In the case of continuous speech recognition, the following advances have converged to make the new technology possible:

- higher-accuracy continuous speech recognition, based on better speech modeling techniques;
- better recognition search strategies that reduce the time needed for high-accuracy recognition; and
- increased power of audio-capable, off-the-shelf workstations.

The paradigm shift is taking place in the way we view and use speech recognition. Rather than being mostly a laboratory endeavor, speech recognition is fast becoming a technology that is pervasive and will have a profound influence on the way humans communicate with machines and with each other.

This paper focuses on speech modeling advances in continuous speech recognition, with an exposition of hidden Markov models (HMMs), the mathematical backbone behind these advances. While knowledge of properties of the speech signal and of speech perception have always played a role, recent improvements have relied largely on solid mathematical and probabilistic modeling methods, especially the use of HMMs for modeling speech sounds. These methods are capable of modeling time and spectral variability simultaneously, and the model parameters can be estimated automatically from given training speech data. The traditional processes of segmentation and labeling of speech sounds are now merged into a single probabilistic process that can optimize recognition accuracy.

This paper describes the speech recognition process and provides typical recognition accuracy figures obtained in laboratory tests as a function of vocabulary, speaker dependence, grammar complexity, and the amount of speech used in training the system. As a result of modeling advances, recognition error rates have dropped several fold. Important to these improvements have been the availability of common speech corpora for training and testing purposes and the adoption of standard testing procedures.

We will argue that future advances in speech recognition must continue to rely on finding better ways to incorporate our speech knowledge into advanced mathematical models, with an emphasis on methods that are robust to speaker variability, noise, and other acoustic distortions.

## THE SPEECH RECOGNITION PROBLEM

Automatic speech recognition can be viewed as a mapping from a continuous-time signal, the speech signal, to a sequence of discrete entities—for example, phonemes (or speech sounds), words, and sentences. The major obstacle to high-accuracy recognition is the large variability in the speech signal characteristics. This variability has three main components: linguistic variability, speaker variability, and channel variability. Linguistic variability includes the effects of phonetics, phonology, syntax, semantics, and discourse on the speech signal. Speaker variability includes intra- and interspeaker variability, including the effects of coarticulation—that is, the effects of neighboring sounds on the acoustic realization of a particular phoneme due to continuity and motion constraints on the human articulatory apparatus. Channel variability includes the effects of background noise and the transmission channel (e.g., microphone, telephone, reverberation). All these variabilities tend to shroud the intended message with layers of uncertainty, which must be unraveled by the recognition process. This paper will focus on modeling linguistic and speaker variabilities for the speech recognition problem.

**Units of Speech.** To gain an appreciation of what modeling is required to perform recognition, we shall use as an example the phrase "grey whales," whose speech signal is shown at the bottom of Fig. 1 with the corresponding spectrogram (or voice print) shown immediately above. The spectrogram shows the result of a frequency analysis of the speech, with the dark bands representing resonances of the vocal tract. At the top of Fig. 1 are the two words "grey" and "whales," which are the desired output of the recognition system. The first thing to note is that the speech signal and the spectrogram show no separation

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

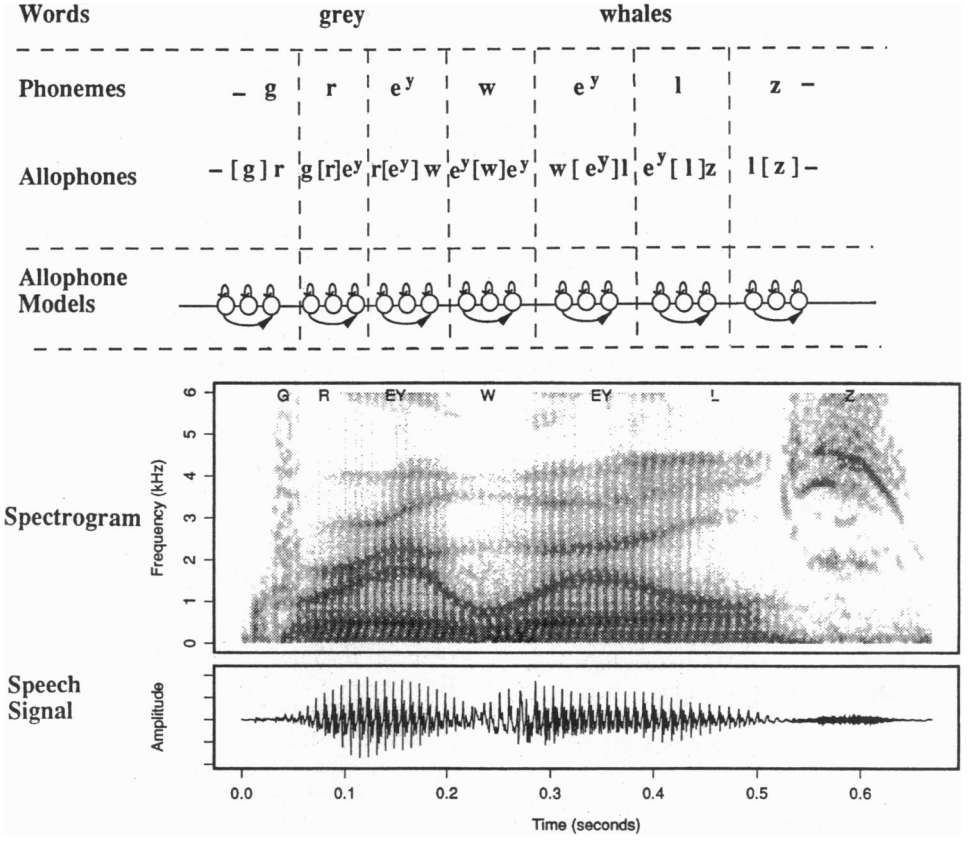


FIG. 1. Units of speech.

between the two words "grey" and "whales" at all; they are in fact connected, as is typical of continuous speech.

Below the word level in Fig. 1 is the phonetic level. Here the words are represented in terms of a phonetic alphabet that tells us what the different sounds in the two words are. In this case the phonetic transcription is given by [g r $e^y$ w $e^y$ l z]. Again, while the sequence of phonemes is discrete, there is no physical separation between the different sounds in the speech signal. In fact, it is not clear where one sound ends and the next begins. The dashed vertical lines shown in Fig. 1 give a rough segmentation of the speech signal, which shows approximately the correspondences between the phonemes and the speech.

Now, the phoneme [$e^y$] occurs once in each of the two words. If we look at the portions of the spectrogram corresponding to the two [$e^y$] phonemes, we notice some similarities between the two parts, but we also note some differences. The differences are mostly due to the fact that the two phonemes are in different contexts: the first [$e^y$] phoneme is preceded by [r] and followed by [w], while the second is preceded by [w] and followed by [l]. These contextual effects are the result of what is known as coarticulation, the fact that the articulation of each sound blends into the articulation of the following sound. In many cases, contextual phonetic effects span several phonemes, but the major effects are caused by the two neighboring phonemes.

To account for the fact that the same phoneme has different acoustic realizations, depending on the context, we refer to each specific context as an allophone. Thus, in Fig. 1, we have two different allophones of the phoneme [$e^y$], one for each of the two contexts in the two words. In this way, we are able to deal with the phonetic variability that is inherent in coarticulation and that is evident in the spectrogram of Fig. 1.

To perform the necessary mapping from the continuous speech signal to the discrete phonetic level, we insert a model—a finite-state machine in our case—for each of the allophones that are encountered. We note from Fig. 1 that the structures of these models are identical; the differences will be in the values given to the various model parameters. Each of these models is a hidden Markov model, which is discussed below.

## HIDDEN MARKOV MODELS

**Markov Chains.** Before we explain what a hidden Markov model is, we remind the reader of what a Markov chain is. A Markov chain consists of a number of states, with transitions among the states. Associated with each transition is a probability and associated with each state is a symbol. Fig. 2 shows a three-state Markov chain, with transition probabilities $a_{ij}$ between states $i$ and $j$. The symbol A is associated with state 1, the symbol B with state 2, and the symbol C with state 3. As one transitions from state 1 to state 2, for example, the symbol B is produced as output. These symbols are called output symbols because a Markov chain is thought of as a generative model; it outputs symbols as one transitions from one state to another. Note that in a Markov chain the transitioning from

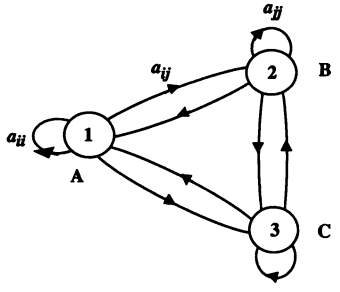


FIG. 2. A three-state Markov chain.

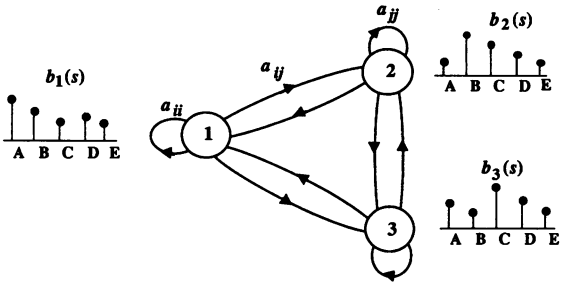


FIG. 3. A three-state HMM.

one state to another is probabilistic, but the production of the output symbols is deterministic.

Now, given a sequence of output symbols that were generated by a Markov chain, one can retrace the corresponding sequence of states completely and unambiguously (provided the output symbol for each state was unique). For example, the sample symbol sequence B A A C B B A C C C A is produced by transitioning into the following sequence of states: 2 1 1 3 2 2 1 3 3 3 1.

**Hidden Markov Models.** A hidden Markov model (HMM) is the same as a Markov chain, except for one important difference: the output symbols in an HMM are probabilistic. Instead of associating a single output symbol per state, in an HMM all symbols are possible at each state, each with its own probability. Thus, associated with each state is a probability distribution of all the output symbols. Furthermore, the number of output symbols can be arbitrary. The different states may then have different probability distributions defined on the set of output symbols. The probabilities associated with states are known as output probabilities.

Fig. 3 shows an example of a three-state HMM. It has the same transition probabilities as the Markov chain of Fig. 2. What is different is that we associate a probability distribution $b_i(s)$ with each state $i$, defined over the set of output symbols $s$—in this case we have five output symbols—A, B, C, D, and E. Now, when we transition from one state to another, the output symbol is chosen according to the probability distribution corresponding to that state. Compared to a Markov chain, the output sequences generated by an HMM are what is known as doubly stochastic: not only is the transitioning from one state to another stochastic (probabilistic) but so is the output symbol generated at each state.

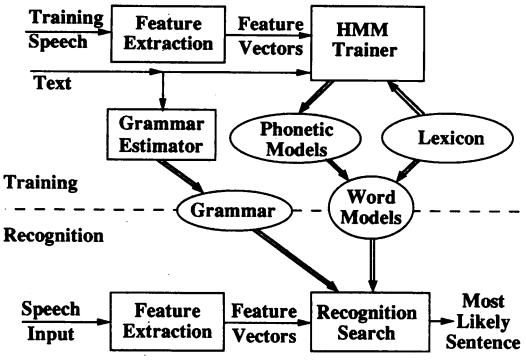


FIG. 5. General system for training and recognition.

Now, given a sequence of symbols generated by a particular HMM, it is not possible to retrace the sequence of states unambiguously. Every sequence of states of the same length as the sequence of symbols is possible, each with a different probability. Given the sample output sequence—C D A A B E D B A C C—there is no way for sure to know which sequence of states produced these output symbols. We say that the sequence of states is hidden in that it is hidden from the observer if all one sees is the output sequence, and that is why these models are known as *hidden* Markov models.

Even though it is not possible to determine for sure what sequence of states produced a particular sequence of symbols, one might be interested in the sequence of states that has the highest probability of having generated the given sequence.

**Phonetic HMMs.** We now explain how HMMs are used to model phonetic speech events. Fig. 4 shows an example of a three-state HMM for a single phoneme. The first stage in the continuous-to-discrete mapping that is required for recognition is performed by the analysis or feature extraction box shown in Fig. 5. Typically, the analysis consists of estimation of the short-term spectrum of the speech signal over a frame (window) of about 20 ms. The spectral computation is then updated about every 10 ms, which corresponds to a frame rate of 100 frames per second. This completes the initial discretization in time. However, the HMM, as depicted in this paper, also requires the definition of a discrete set of "output symbols." So, we need to discretize the spectrum into one of a finite set of spectra. Fig. 4 depicts a set of spectral templates (known as a codebook) that represent the space of possible

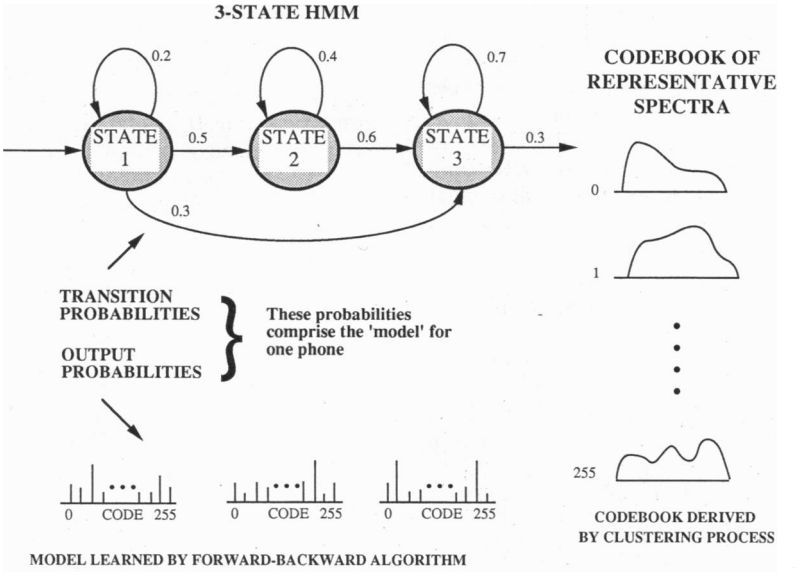


FIG. 4. Basic structure of a phonetic HMM.

speech spectra. Given a computed spectrum for a frame of speech, one can find the template in the codebook that is "closest" to that spectrum, using a process known as vector quantization (1). The size of the codebook in Fig. 4 is 256 templates. These templates, or their indices (from 0 to 255), serve as the output symbols of the HMM. We see in Fig. 4 that associated with each state is a probability distribution on the set of 256 symbols. The definition of a phonetic HMM is now complete. We now describe how it functions.

Let us first see how a phonetic HMM functions as a generative (synthesis) model. As we enter into state 1 in Fig. 4, one of the 256 output symbols is generated based on the probability distribution corresponding to state 1. Then, based on the transition probabilities out of state 1, a transition is made either back to state 1 itself, to state 2, or to state 3, and another symbol is generated based on the probability distribution corresponding to the state into which the transition is made. In this way a sequence of symbols is generated until a transition out of state 3 is made. At that point, the sequence corresponds to a single phoneme.

The same model can be used in recognition mode. In this mode each model can be used to compute the probability of having generated a sequence of spectra. Assuming we start with state 1 and given an input speech spectrum that has been quantized to one of the 256 templates, one can perform a table lookup to find the probability of that spectrum. If we now assume that a transition is made from state 1 to state 2, for example, the previous output probability is multiplied by the transition probability from state 1 to state 2 (0.5 in Fig. 4). A new spectrum is now computed over the next frame of speech and quantized; the corresponding output probability is then determined from the output probability distribution corresponding to state 2. That probability is multiplied by the previous product, and the process is continued until the model is exited. The result of multiplying the sequence of output and transition probabilities gives the total probability that the input spectral sequence was "generated" by that HMM using a specific sequence of states. For every sequence of states, a different probability value results. For recognition, the probability computation just described is performed for all possible phoneme models and all possible state sequences. The one sequence that results in the highest probability is declared to be the recognized sequence of phonemes.

We note in Fig. 4 that not all transitions are allowed (i.e., the transitions that do not appear have a probability of zero). This model is what is known as a "left-to-right" model, which represents the fact that, in speech, time flows in a forward direction only; that forward direction is represented in Fig. 4 by a general left-to-right movement. Thus, there are no transitions allowed from right to left. Transitions from any state back to itself serve to model variability in time, which is very necessary for speech since different instantiations of phonemes and words are uttered with different time registrations. The transition from state 1 to state 3 means that the shortest phoneme that is modeled by the model in Fig. 4 is one that is two frames long, or 20 ms. Such a phoneme would occupy state 1 for one frame and state 3 for one frame only. One explanation for the need for three states, in general, is that state 1 corresponds roughly to the left part of the phoneme, state 2 to the middle part, and state 3 to the right part. (More states can be used, but then more data would be needed to estimate their parameters robustly.)

Usually, there is one HMM for each of the phonetic contexts of interest. Although the different contexts could have different structures, usually all such models have the same structure as the one shown in Fig. 4; what makes them different are the transition and output probabilities.

## A HISTORICAL OVERVIEW

HMM theory was developed in the late 1960s by Baum and Eagon (2) at the Institute for Defense Analyses (IDA). Initial work using HMMs for speech recognition was performed in the 1970s at IDA, IBM (3), and Carnegie–Mellon University (4). In 1980 a number of researchers in speech recognition in the United States were invited to a workshop in which IDA researchers reviewed the properties of HMMs and their use for speech recognition. That workshop prompted a few organizations, such as AT&T and BBN, to start working with HMMs (5, 6). In 1984 a program in continuous speech recognition was initiated by the Advanced Research Projects Agency (ARPA), and soon thereafter HMMs were shown to be superior to other approaches (7). Until then, only a handful of organizations worldwide had been working with HMMs. Because of the success of HMMs and because of the strong influence of the ARPA program, with its emphasis on periodic evaluations using common speech corpora, the use of HMMs for speech recognition started to spread worldwide. Today, their use has dominated other approaches to speech recognition in dozens of laboratories around the globe. In addition to the laboratories mentioned above, significant work is taking place at, for example, the Massachusetts Institute of Technology's Lincoln Laboratory, Dragon, SRI, and TI in the United States; CRIM and BNR in Canada; RSRE and Cambridge University in the United Kingdom; ATR, NTT, and NEC in Japan; LIMSI in France; Philips in Germany and Belgium; and CSELT in Italy, to name a few. Comprehensive treatments of HMMs and their utility in speech recognition can be found in Rabiner (8), Lee (9), Huang *et al.* (10), Rabiner and Juang (11), and the references therein. Research results in this area are usually reported in the following journals and conference proceedings: *IEEE Transactions on Speech and Audio Processing*; *IEEE Transactions on Signal Processing; Speech Communication Journal*; IEEE International Conference on Acoustics, Speech, and Signal Processing; EuroSpeech; and the International Conference on Speech and Language Processing.

HMMs have proven to be a good model of speech variability in time and feature space. The automatic training of the models from speech data has accelerated the speed of research and improved recognition performance. Also, the probabilistic formulation of HMMs has provided a unified framework for scoring of hypotheses and for combining different knowledge sources. For example, the sequence of spoken words can also be modeled as the output of another statistical process (12). In this way it becomes natural to combine the HMMs for speech with the statistical models for language.

## TRAINING AND RECOGNITION

Fig. 5 shows a block diagram of a general system for training and recognition. Note that in both training and recognition the first step in the process is to perform feature extraction on the speech signal.

**Feature Extraction.** In theory it should be possible to recognize speech directly from the signal. However, because of the large variability of the speech signal, it is a good idea to perform some form of feature extraction to reduce that variability. In particular, computing the envelope of the short-term spectrum reduces the variability significantly by smoothing the detailed spectrum, thus eliminating various source characteristics, such as whether the sound is voiced or fricated, and, if voiced, it eliminates the effect of the periodicity or pitch. The loss of source information does not appear to affect recognition performance much because it turns out that the spectral envelope is highly correlated with the source information.

One reason for computing the short-term spectrum is that the cochlea of the human ear performs a quasi-frequency analysis. The analysis in the cochlea takes place on a nonlinear

frequency scale (known as the Bark scale or the mel scale). This scale is approximately linear up to about 1000 Hz and is approximately logarithmic thereafter. So, in the feature extraction, it is very common to perform a frequency warping of the frequency axis after the spectral computation.

Researchers have experimented with many different types of features for use with HMMs (11). Variations on the basic spectral computation, such as the inclusion of time and frequency masking, have been shown to provide some benefit in certain cases. The use of auditory models as the basis for feature extraction has been useful in some systems (13), especially in noisy environments (14).

Perhaps the most popular features used for speech recognition with HMMs today are what are known as mel-frequency cepstral coefficients or MFCCs (15). After the mel-scale warping of the spectrum, the logarithm of the spectrum is taken and an inverse Fourier transform results in the cepstrum. By retaining the first dozen or so coefficients of the cepstrum, one would be retaining the spectral envelope information that is desired. The resulting features are the MFCCs, which are treated as a single vector and are typically computed for every frame of 10 ms. These feature vectors form the input to the training and recognition systems.

**Training.** Training is the process of estimating the speech model parameters from actual speech data. In preparation for training, what is needed is the text of the training speech and a lexicon of all the words in the training, along with their pronunciations, written down as phonetic spellings. Thus, a transcription of the training speech is made by listening to the speech and writing down the sequence of words. All the distinct words are then placed in a lexicon and someone has to provide a phonetic spelling of each word. In cases where a word has more than one pronunciation, as many phonetic spellings as there are pronunciations are included for each word. These phonetic spellings can be obtained from existing dictionaries or they can be written by anyone with minimal training in phonetics.

*Phonetic HMMs and lexicon.* Given the training speech, the text of the speech, and the lexicon of phonetic spellings of all the words, the parameters of all the phonetic HMMs (transition and output probabilities) are estimated automatically using an iterative procedure known as the Baum-Welch or forward-backward algorithm (2). This algorithm estimates the parameters of the HMMs so as to maximize the likelihood (probability) that the training speech was indeed produced by these HMMs. The iterative procedure is guaranteed to converge to a local optimum. Typically, about five iterations through the data are needed to obtain a reasonably good estimate of the speech model. [See the paper by Jelinek (16) for more details on the HMM training algorithm.]

It is important to emphasize the fact that HMM training does not require that the data be labeled in detail in terms of the location of the different words and phonemes; that is, no time alignment between the speech and the text is needed. Given a reasonable initial estimate of the HMM parameters, the Baum-Welch training algorithm performs an implicit alignment of the input spectral sequence to the states of the HMM, which is then used to obtain an improved estimate. All that is required in addition to the training speech is the text transcription and the lexicon. This is one of the most important properties of the HMM approach to recognition. Training does require significant amounts of computing but does not require much in terms of human labor.

In preparation for recognition it is important that the lexicon contain words that would be expected to occur in future data, even if they did not occur in the training. Typically, closed-set word classes are filled out—for example, days of the week, months of the year, numbers.

After completing the lexicon, HMM word models are compiled from the set of phonetic models using the phonetic spellings in the lexicon. These word models are simply a concatenation of the appropriate phonetic HMM models. We then compile the grammar (which specifies sequences of words) and the lexicon (which specifies sequences of phonemes for each word) into a single probabilistic grammar for the sequences of phonemes. The result of the recognition is a particular sequence of words, corresponding to the recognized sequence of phonemes.

*Grammar.* Another aspect of the training that is needed to aid in the recognition is to produce the grammar to be used in the recognition. Without a grammar, all words would be considered equally likely at each point in an utterance, which would make recognition difficult, especially with large vocabularies. We, as humans, make enormous use of our knowledge of the language to help us recognize what a person is saying. A grammar places constraints on the sequences of the words that are allowed, giving the recognition fewer choices at each point in the utterance and, therefore, improving recognition performance.

Most grammars used in speech recognition these days are statistical Markov grammars that give the probabilities of different sequences of words—so-called $n$-gram grammars. For example, bigram grammars give the probabilities of all pairs of words, while trigram grammars give the probabilities of all triplets of words in the lexicon. In practice, trigrams appear to be sufficient to embody much of the natural constraints imposed on the sequences of words in a language. In an $n$-gram Markov grammar, the probability of a word is a function of the previous $n - 1$ words. While this assumption may not be valid in general, it appears to be sufficient to result in good recognition accuracy. Furthermore, the assumption allows for efficient computation of the likelihood of a sequence of words.

A measure of how constrained a grammar is is given by its *perplexity* (12). Perplexity is defined as 2 raised to the power of the Shannon entropy of the grammar. If all words are equally likely at each point in a sentence, the perplexity is equal to the vocabulary size. In practice, sequences of words have greatly differing probabilities, and the perplexity is often much less than the vocabulary size, especially for larger vocabularies. Because grammars are estimated from a set of training data, it is often more meaningful to measure the perplexity on an independent set of data, or what is known as test-set perplexity (12). Test-set perplexity $Q$ is obtained by computing

$$Q = P(w_1\, w_2 \ldots w_M)^{-1/M}$$

where $w_1\, w_2 \ldots w_M$ is the sequence of words obtained by concatenating all the test sentences and $P$ is the probability of that whole sequence. Because of the Markov property of $n$-gram grammars, the probability $P$ can be computed as the product of consecutive conditional probabilities of $n$-grams.

**Recognition.** As shown in Fig. 5, the recognition process starts with the feature extraction stage, which is identical to that performed in the training. Then, given the sequence of feature vectors, the word HMM models, and the grammar, the recognition is simply a large search among all possible word sequences for that word sequence with the highest probability to have generated the computed sequence of feature vectors. In theory the search is exponential with the number of words in the utterance. However, because of the Markovian property of conditional independence in the HMM, it is possible to reduce the search drastically by the use of dynamic programming using, for example, the Viterbi algorithm (17). The Viterbi algorithm requires computation that is proportional to the number of states in the model and the length of the input sequence. Further approximate search algorithms have been developed that allow the search computation to be reduced further, without significant loss in performance. The most commonly used technique is the beam search (18), which avoids the computation for states that have low probability.

## STATE OF THE ART

In this section we review the state of the art in continuous speech recognition. We present some of the major factors that led to the relatively large improvements in performance and give sample performance figures under different conditions. We then review several of the issues that affect performance, including the effects of training and grammar, speaker-dependent versus speaker-independent recognition, speaker adaptation, nonnative speakers, and the inclusion of new words in the speech. Most of the results and examples below have been taken from the ARPA program, which has sponsored the collection and dissemination of large speech corpora for comparative evaluation.

**Improvements in Performance.** The improvements in speech recognition performance have been so dramatic that in the ARPA program the word error rate has dropped by a factor of 5 in 5 years! This unprecedented advance in the state of the art is due to four factors: use of common speech corpora, improved acoustic modeling, improved language modeling, and a faster research experimentation cycle.

*Common speech corpora.* The ARPA program must be given credit for starting and maintaining a sizable program in large-vocabulary, speaker-independent, continuous speech recognition. One of the cornerstones of the ARPA program has been the collection and use of common speech corpora for system development and testing. (The various speech corpora collected under this program are available from the Linguistic Data Consortium, with offices at the University of Pennsylvania.) Through cycles of algorithm development, evaluation, and sharing of detailed technical information, work in the program led to the incredible reduction in error rate noted above.

*Acoustic modeling.* A number of ideas in acoustic modeling have led to significant improvements in performance. Developing HMM phonetic models that depend on context—that is, on the left and right phonemes—have been shown to reduce the word error rate by about a factor of 2 over context-independent models (7). One of the properties of HMMs is that different models (e.g., context-independent, diphone, and triphone models) can be interpolated in such a way as to make the best possible use of the training data, thus increasing the robustness of the system.

The modeling of cross-word effects is also important, especially for small words, such as function words (where many of the errors occur), and can reduce the overall word error rate by about 20 percent.

In addition to the use of feature vectors, such as MFCCs, it has been found that including what is known as delta features—the change in the feature vector over time—can reduce the error rate by a factor of about 2 (19). The delta features are treated like an additional feature vector whose probability distribution must also be estimated from training data. Even though the original feature vector contains all the information that can be used for the recognition, it appears that the HMM does not take full advantage of the time evolution of the feature vectors. Computing the delta parameters is a way of extracting that time information and providing it to the HMM directly (20).

Proper estimation of the HMM parameters—the transition and output probabilities—from training data is of crucial importance. Because only a small number of the possible feature vector values will occur in any training set, it is important to use probability estimation and smoothing techniques that not only will model the training data well but also will model other possible occurrences in future unseen data. A number of probability estimation and smoothing techniques have been developed that strike a good compromise between computation, robustness, and recognition accuracy and have resulted in error rate reductions of about 20 percent compared to the discrete HMMs presented in this paper (10, 21–23).

*Language modeling.* Statistical $n$-gram grammars, especially word trigrams, have been very successful in modeling the likely word sequences in actual speech data. To obtain a good language model, it is important to use as large a text corpus as possible so that all the trigrams to be seen in any new test material are seen in the training with about the same probability. Note that only the text is needed for training the language model, not the actual speech. Typically, millions of words of text are used to develop good language models. A number of methods have been developed that provide a robust estimate of the trigram probabilities (24, 25).

*Research experimentation cycle.* We have emphasized above the recognition improvements that have been possible with innovations in algorithm development. However, those improvements would not have been possible without the proper computational tools that have allowed the researcher to shorten the research experimentation cycle. Faster search algorithms, as well as faster workstations, have made it possible to run a large experiment in a short time, typically overnight, so that the researcher can make appropriate changes the next day and run another experiment. The combined increases in speed with better search and faster machines have been several orders of magnitude.

**Sample Performance Figures.** Fig. 6 gives a representative sampling of state-of-the-art continuous speech recognition performance. The performance is shown in terms of the word error rate, which is defined as the sum of word substitutions,

| Corpus | Training Data | | Vocabulary | | Test Data | | Word Error Rate |
|---|---|---|---|---|---|---|---|
| | Type | Amount | Size | Open/ Closed | Type | Perplexity | |
| TI Digits | Read | 4 hrs | 10 | Closed | Read | 11 | 0.3% |
| ARPA Resource Management | Read | 4 hrs | 1000 | Closed | Read | 60 | 4% |
| ARPA Airline Travel | Spontaneous | 13 hrs | 1800 | Open | Spontaneous | 12 | 4% |
| ARPA Wall Street Journal Dictation | Read | 12 hrs | 5000 | Closed | Read | 45 | 5% |
| | Read | 12 hrs | 20,000 | Open | Read | 200 | 13% |
| | Read | 12 hrs | 20,000 | Open | Spontaneous | 255 | 26% |

FIG. 6. State of the art in speaker-independent, continuous speech recognition.

deletions, and insertions, as a percentage of the actual number of words in the test. All training and test speakers were native speakers of American English. The error rates are for speaker-independent recognition; that is, test speakers were different from the speakers used for training. All the results in Fig. 6 are for laboratory systems; they were obtained from refs. 26–30.

The results for four corpora are shown: the TI connected-digit corpus (31), the ARPA Resource Management corpus (32), the ARPA Airline Travel Information Service (ATIS) corpus (33), and the ARPA Wall Street Journal (WSJ) corpus (34). The first two corpora were collected in very quiet rooms at TI, while the latter two were collected in office environments at several different sites. The ATIS corpus was collected from subjects trying to access airline information by voice using natural English queries; it is the only corpus of the four presented here for which the training and test speech are spontaneous instead of being read sentences. The WSJ corpus consists largely of read sentences from the *Wall Street Journal*, with some spontaneous sentences used for testing. Shown in Fig. 6 are the vocabulary size for each corpus and whether the vocabulary is closed or open. The vocabulary is closed when all the words in the test are guaranteed to be in the system's lexicon, while in the open condition the test may contain words that are not in the system's lexicon and, therefore, will cause errors in the recognition. The perplexity is the test-set perplexity defined above. Strictly speaking, perplexity is not defined for the open-vocabulary condition, so the value of the perplexity that is shown was obtained by making some simple assumptions about the probability of $n$-grams that contain the unknown words.

The results shown in Fig. 6 are average results over a number of test speakers. The error rates for individual speakers vary over a relatively wide range and may be several times lower or higher than the average values shown. Since much of the data were collected in relatively benign conditions, one would expect the performance to degrade in the presence of noise and channel distortion. It is clear from Fig. 6 that higher perplexity, open vocabulary, and spontaneous speech tend to increase the word error rate. We shall quantify some of these effects next and discuss some important issues that affect performance.

**Effects of Training and Grammar.** It is well recognized that increasing the amount of training data generally decreases the word error rate. However, it is important that the increased training be representative of the types of data in the test. Otherwise, the increased training might not help.

With the RM corpus, it has been found that the error rate is inversely proportional to the square root of the amount of training data, so that quadrupling the training data results in cutting the word error rate by a factor of 2. This large reduction in error rate by increasing the training data may have been the result of an artifact of the RM corpus; namely, that the sentence patterns of the test data were the same as those in the training. In a realistic corpus, where the sentence patterns of the test can often be quite different from the training, such improvements may not be as dramatic. For example, recent experiments with the WSJ corpus have failed to show significant reduction in error rate by doubling the amount of training. However, it is possible that increasing the complexity of the models as the training data are increased could result in larger reduction in the error rate. This is still very much a research issue.

Word error rates generally increase with an increase in grammar perplexity. A general rule of thumb is that the error rate increases as the square root of perplexity, with everything else being equal. This rule of thumb may not always be a good predictor of performance, but it is a reasonable approximation. Note that the size of the vocabulary as such is not the primary determiner of recognition performance but rather the freedom in which the words are put together, which is represented by the grammar. A less constrained grammar, such as in the WSJ corpus, results in higher error rates.

**Speaker-Dependent vs. Speaker-Independent Recognition.** The terms speaker-dependent (SD) and speaker-independent (SI) recognition are often used to describe different modes of operation of a speech recognition system. SD recognition refers to the case when a single speaker is used to train the system and the same speaker is used to test the system. SI recognition refers to the case where the test speaker is not included in the training. HMM-based systems can operate in either SD or SI mode, depending on the training data used. In SD mode training speech is collected from a single speaker only, while in SI mode training speech is collected from a variety of speakers.

SD and SI modes of recognition can be compared in terms of the word error rates for a given amount of training. A general rule of thumb is that, if the total amount of training speech is fixed at some level, the SI word error rates are about four times the SD error rates. Another way of stating this rule of thumb is that, for SI recognition to have the same performance as SD recognition, requires about 15 times the amount of training data (35). These results were obtained when 1 hr of speech was used to compute the SD models. However, in the limit, as the amount of training speech for SD and SI models is made larger and larger, it is not clear that any amount of training data will allow SI performance to approach SD performance.

**Adaptation.** It is possible to improve the performance of an SI system by incrementally adapting to the voice of a new speaker as the speaker uses the system. This would be especially needed for atypical speakers with high error rates who might otherwise find the system unusable. Such speakers would include speakers with unusual dialects and those for whom the SI models simply are not good models of their speech. However, incremental adaptation could require hours of usage and a lot of patience from the new user before the performance becomes adequate.

A good solution to the atypical speaker problem is to use a method known as rapid speaker adaptation. In this method only a small amount of speech (about 2 min) is collected from the new speaker before using the system. By having the same utterances collected previously from one or more prototype speakers, methods have been developed for deriving a speech model for the new speaker through a simple transformation on the speech model of the prototype speakers (36–38). It is possible with these methods to achieve average SI performance for speakers who otherwise would have several times the error rate.

**Out-of-Vocabulary Words.** Out-of-vocabulary words cause recognition errors and degrade performance. There have been very few attempts at automatically detecting the presence of new words, with limited success (39). Most systems simply do not do anything special to deal with the presence of such words. Experiments have shown that, if the new words are added to the system's lexicon but without additional training for the new words, the SI error rate for the new words is about twice that with training that includes the new words. Therefore, user-specified vocabulary and grammar can be easily incorporated into a speech recognition system at a modest increase in the error rate for the new words.

## REAL-TIME SPEECH RECOGNITION

Until recently, it was thought that to perform high-accuracy, real-time, continuous speech recognition for large vocabularies would require either special-purpose VLSI hardware or a multiprocessor. However, new developments in search algorithms have sped up the recognition computation at least two orders of magnitude, with little or no loss in recognition accuracy (40–44). In addition, computing advances have achieved two-orders-magnitude increase in workstation

speeds in the past decade. These two advances have made software-based, real-time, continuous speech recognition a reality. The only requirement is that the workstation must have an A/D converter to digitize the speech. All the signal processing, feature extraction, and recognition search is then performed in software in real time on a single-processor workstation.

For example, it is now possible to perform a 2000-word ATIS task in real time on workstations such as the Silicon Graphics Indigo R3000 or the Sun SparcStation 2. Most recently, a 20,000-word WSJ continuous dictation task was demonstrated in real time (45) on a Hewlett–Packard 735 workstation, which has about three times the power of an SGI R3000. Thus, the computation grows much slower than linear with the size of the vocabulary.

The real-time feats just described have been achieved at a relatively small cost in word accuracy. Typically, the word error rates are less than twice those of the best research systems.

## CONCLUDING REMARKS

We are on the verge of an explosion in the integration of speech recognition in a large number of applications. The ability to perform software-based, real-time recognition on a workstation will no doubt change the way people think about speech recognition. Anyone with a workstation can now have this capability on their desk. In a few years, speech recognition will be ubiquitous and will enter many aspects of our lives. This paper reviewed the technologies that made these advances possible.

Despite all these advances, much remains to be done. Speech recognition performance for very large vocabularies and larger perplexities is not yet adequate for useful applications, even under benign acoustic conditions. Any degradation in the environment or changes between training and test conditions causes a degradation in performance. Therefore, work must continue to improve robustness to varying conditions: new speakers, new dialects, different channels (microphones, telephone), noisy environments, and new domains and vocabularies. What will be especially needed are improved mathematical models of speech and language and methods for fast adaptation to new conditions.


1. Makhoul, J., Roucos, S. & Gish, H. (1985) *Proc. IEEE* **73,** 1551–1588.
2. Baum, L. E. & Eagon, J. A. (1967) *Am. Math. Soc. Bull.* **73,** 360–362.
3. Jelinek, F., Bahl, L. R. & Mercer, R. L. (1975) *IEEE Trans. Inf. Theory* **21,** 250–256.
4. Baker, J. K. (1975) in *Speech Recognition*, ed. Reddy, R. (Academic, New York), pp. 521–542.
5. Levinson, S. E., Rabiner, L. R. & Sondhi, M. M. (1983) *Bell Syst. Tech. J.* **62,** 1035–1073.
6. Schwartz, R. M., Chow, Y., Roucos, S., Krasner, M. & Makhoul, J. (1984) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (San Diego), pp. 35.6.1–35.6.4.
7. Chow, Y. L., Schwartz, R. M., Roucos, S., Kimball, O. A., Price, P. J., Kubala, G. F., Dunham, M. O., Krasner, M. A. & Makhoul, J. (1986) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Tokyo), pp. 1593–1596.
8. Rabiner, L. R. (1989) *Proc. IEEE* **77,** 257–286.
9. Lee, K.-F. (1989) *Automatic Speech Recognition: The Development of the Sphinx System* (Kluwer, Boston).
10. Huang, X. D., Ariki, Y. & Jack, M. A. (1990) *Hidden Markov Models for Speech Recognition* (Edinburgh Univ. Press, Edinburgh).
11. Rabiner, L. R. & Juang, B.-H. (1993) *Fundamentals of Speech Recognition* (Prentice–Hall, Englewood Cliffs, NJ).
12. Bahl, L. R., Jelinek, F. & Mercer, R. L. (1983) *IEEE Trans. Pat. Anal. Mach. Intell.* **PAMI-5,** 179–190.
13. Cohen, J. (1989) *J. Acoust. Soc. Am.* **85,** 2623–2629.
14. Hunt, M., Richardson, S., Bateman, D. & Piau, A. (1991) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Toronto), pp. 881–884.
15. Davis, S. & Mermelstein, P. (1980) *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-28,** 357–366.
16. Jelinek, F. (1995) *Proc. Natl. Acad. Sci. USA* **92,** 9964–9969.
17. Forney, G. D. (1973) *Proc. IEEE* **61,** 268–278.
18. Lowerre, B. T. (1976) Doctoral thesis (Carnegie–Mellon Univ., Pittsburgh).
19. Furui, S. (1986) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Tokyo), pp. 1991–1994.
20. Gupta, V. N., Lennig, M. & Mermelstein, P. (1987) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Dallas), pp. 697–700.
21. Bellegarda. J. R. & Nahamoo, D. H. (1989) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Glasgow, Scotland), pp. 13–16.
22. Gauvain, J. L. & Lee, C. H. (1992) *Speech Commun.* **11,** Nos. 2 and 3.
23. Schwartz, R., Kimball, O., Kubala, F., Feng, M., Chow, Y., Barry, C. & Makhoul, J. (1989) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Glasgow, Scotland), Paper S10b.9.
24. Katz, S. (1987) *IEEE Trans. Acoust. Speech Signal Process.* **35,** 400–401.
25. Placeway, P., Schwartz, R., Fung, P. & Nguyen, L. (1993) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Minneapolis), pp. II-33–II-36.
26. Bates, M., Bobrow, R., Fung, P., Ingria, R., Kubala, F., Makhoul, J., Nguyen, L., Schwartz, R. & Stallard, D. (1993) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Minneapolis), pp. II-111–II-114.
27. Cardin, R., Normandin, Y. & Millien, E. (1993) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Minneapolis), pp. II-243–II-246.
28. Haeb-Umbach, R., Geller, D. & Ney, H. (1993) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Minneapolis), pp. II-239–II-242.
29. Huang, X. D., Lee, K. F., Hon, H. W. & Hwang, M.-Y. (1991) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Toronto), Vol. S1, pp. 345–347.
30. Pallett, D., Fiscus, J., Fisher, W. & Garofolo, J. (1993) in *Proceedings of the ARPA Workshop on Human Language Technology* (Plainsboro, NJ), (Kaufmann, San Francisco), pp. 7–18.
31. Leonard, R. G. (1984) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (San Diego), Paper 42.11.
32. Price, P., Fisher, W. M., Bernstein, J. & Pallett, D. S. (1988) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (New York), pp. 651–654.
33. MADCOW (1992) in *Proceedings of the DARPA Speech and Natural Language Workshop* (Harriman, NY), (Kaufmann, San Mateo, CA), pp. 7–14.
34. Paul, D. (1992) in *Proceedings of the DARPA Speech and Natural Language Workshop* (Harriman, NY), (Kaufmann, San Mateo, CA), pp. 357–360.
35. Schwartz, R., Anastasakos, A., Kubala, F., Makhoul, J., Nguyen, L. & Zavaliagkos, G. (1993) in *Proceedings of the ARPA Workshop on Human Language Technology* (Plainsboro, NJ), (Kaufmann, San Francisco), pp. 75–80.
36. Furui, S. (1989) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Glasgow, Scotland), Paper S6.9.
37. Kubala, F. & Schwartz, R. (1990) in *International Conference on Speech and Language Processing* (Kobe, Japan), pp. 153–156.
38. Nakamura, S. & Shikano, K. (1989) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Glasgow, Scotland), Paper S3.3.
39. Asadi, A., Schwartz, R. & Makhoul, J. (1990) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Albuquerque), pp. 125–128.
40. Austin, S., Schwartz, R. & Placeway, P. (1991) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Toronto), pp. 697–700.
41. Bahl, L. R., de Souza, P., Gopalakrishnan, P. S., Kanevsky, D. & Nahamoo, D. (1990) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Albuquerque), pp. 85–88.
42. Ney, H. (1992) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (San Francisco), pp. I-9–I-12.
43. Schwartz, R. & Austin, S. (1991) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Toronto), pp. 701–704.
44. Soong, F. & Huang, E. (1991) in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Toronto), pp. 705–708.
45. Nguyen, L., Schwartz, R., Kubala, F. & Placeway, P. (1993) in *Proceedings of the ARPA Workshop on Human Language Technology* (Plainsboro, NJ), (Kaufmann, San Francisco), pp. 91–95.