

# 15

## An Overview of Modern Speech Recognition

15.1	Introduction .....	339
15.2	Major Architectural Components .....	340
	Acoustic Models • Language Models • Decoding	
15.3	Major Historical Developments in Speech Recognition .....	350
15.4	Speech-Recognition Applications .....	352
	IVR Applications • Appliance—“Response Point” • Mobile Applications	
15.5	Technical Challenges and Future Research Directions .....	356
	Robustness against Acoustic Environments and a Multitude of Other Factors • Capitalizing on Data Deluge for Speech Recognition • Self-Learning and Adaptation for Speech Recognition • Developing Speech Recognizers beyond the Language Barrier • Detection of Unknown Events in Speech Recognition • Learning from Human Speech Perception and Production • Capitalizing on New Trends in Computational Architectures for Speech Recognition • Embedding Knowledge and Parallelism into Speech-Recognition Decoding	
15.6	Summary .....	363
	References.....	363

Xuedong Huang and  
Li Deng  
*Microsoft Corporation*

### 15.1 Introduction

The task of speech recognition is to convert speech into a sequence of words by a computer program. As the most natural communication modality for humans, the ultimate dream of speech recognition is to enable people to communicate more naturally and effectively. While the long-term objective requires deep integration with many NLP components discussed in this book, there are many emerging applications that can be readily deployed with the core speech-recognition module we review in this chapter. Some of these typical applications include voice dialing, call routing, data entry and dictation, command and control, and computer-aided language learning. Most of these modern systems are typically based on statistic models such as hidden Markov models (HMMs). One reason why HMMs are popular is that their parameters can be estimated automatically from a large amount of data, and they are simple and computationally feasible.

Speech recognition is often regarded as the front-end for many NLP components discussed in this book. In practice, the speech system typically uses context-free grammar (CFG) or statistic n-grams for the same reason that HMMs are used for acoustic modeling. There are a number of excellent books that have covered the basis of speech recognition and related spoken language-processing technologies (Lee, 1988; Rabiner and Juang, 1993; Lee et al., 1996; Jelinek, 1997; Gold and Morgan, 2000; Jurafsky and Martin, 2000; O’Shaughnessy, 2000; Huang et al., 2001; Furui, 2001; Deng and O’Shaughnessy, 2003).

AQ1

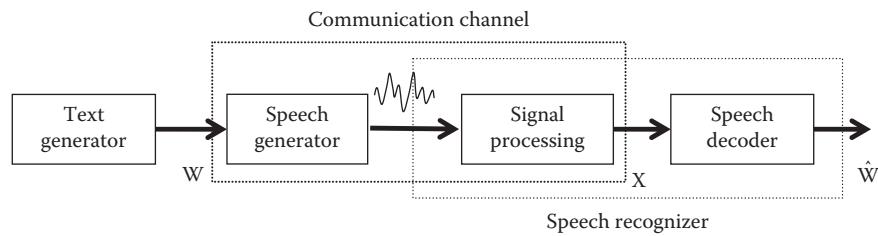
339

In this chapter, we provide an overview in Section 15.2 of the main components in speech recognition, followed by a critical review of the historically significant developments in the field in Section 15.3. We devote Section 15.4 to speech-recognition applications, including some recent case studies. An in-depth analysis of the current state of speech recognition and detailed discussions on a number of future research directions in speech recognition are presented in Section 15.5.

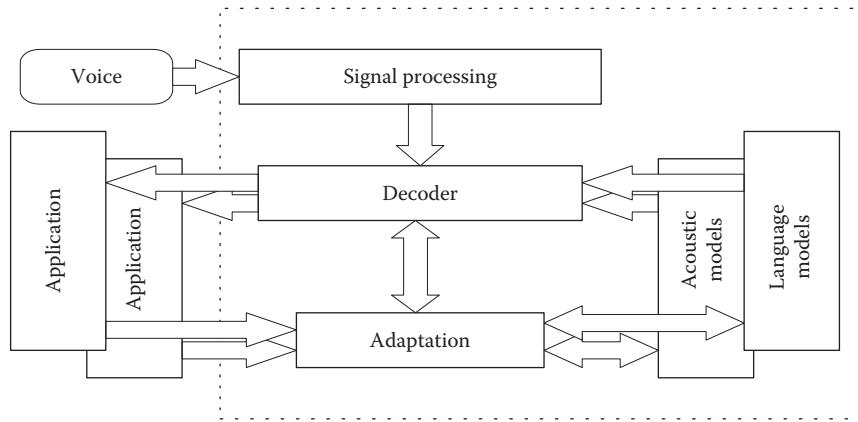
## 15.2 Major Architectural Components

Modern speech-recognition systems have been built invariably based on statistical principles, as pioneered by the work of Baker (1975) and Jelinek (1976) and exposed in detail in Huang et al. (2001). A source-channel mathematical model or a type of generative statistical model is often used to formulate speech-recognition problems. As illustrated in Figure 15.1, the speaker’s mind decides the source word sequence  $W$  that is delivered through his or her text generator. The source is passed through a noisy communication channel that consists of the speaker’s vocal apparatus to produce the speech waveform and the speech signal-processing component of the speech recognizer. Finally, the speech decoder aims to decode the acoustic signal  $X$  into a word sequence  $\hat{W}$ , which is in ideal cases close to the original word sequence  $W$ .

A typical, practical speech-recognition system consists of basic components shown in the dotted box of Figure 15.2. Applications interface with the decoder to obtain recognition results that may be used to adapt other components in the system. *Acoustic models* include the representation of knowledge about acoustics, phonetics, microphone and environment variability, gender and dialect differences among speakers, etc. *Language models* refer to a system’s knowledge of what constitutes a possible word, what words are likely to co-occur, and in what sequence. The semantics and functions related to an operation a user may wish to



**FIGURE 15.1** A source-channel model for a typical speech-recognition system.



**FIGURE 15.2** Basic system architecture of a speech-recognition system.

perform may also be necessary for the language model. Many uncertainties exist in these areas, associated with speaker characteristics, speech style and rate, the recognition of basic speech segments, possible words, likely words, unknown words, grammatical variation, noise interference, nonnative accents, and the confidence scoring of results. A successful speech-recognition system must contend with all of these uncertainties. The acoustic uncertainty of the different accents and speaking styles of individual speakers are compounded by the lexical and grammatical complexity and variations of spoken language, which are all represented in the language model.

As shown in Figure 15.2, the speech signal is processed in the signal-processing module that extracts salient feature vectors for the decoder. The decoder uses both acoustic and language models to generate the word sequence that has the maximum posterior probability for the input feature vectors. It can also provide information needed for the adaptation component to modify either the acoustic or language models so that improved performance can be obtained.

The division of acoustic modeling and language modeling discussed above can be succinctly described by the fundamental equation of statistical speech recognition:

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{w}} P(\mathbf{W}|\mathbf{A}) = \arg \max_{\mathbf{w}} \frac{P(\mathbf{W})P(\mathbf{A}|\mathbf{W})}{P(\mathbf{A})} \quad (15.1)$$

where for the given acoustic observation or feature vector sequence  $\mathbf{X} = X_1X_2 \dots X_n$ , the goal of speech recognition is to find out the corresponding word sequence  $\hat{\mathbf{W}} = w_1w_2 \dots w_m$  that has the maximum posterior probability  $P(\mathbf{W}|\mathbf{X})$  as expressed with Equation 15.1. Since the maximization of Equation 15.1 is carried out with the observation  $\mathbf{X}$  fixed, the above maximization is equivalent of the maximization of the numerator:

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{w}} P(\mathbf{W})P(\mathbf{X}|\mathbf{W}) \quad (15.2)$$

where  $P(\mathbf{W})$  and  $P(\mathbf{X}|\mathbf{W})$  constitute the probabilistic quantities computed by the language modeling and acoustic modeling components, respectively, of speech-recognition systems.

The practical challenge is how to build accurate acoustic models,  $P(\mathbf{X}|\mathbf{W})$ , and language models,  $P(\mathbf{W})$ , which can truly reflect the spoken language to be recognized. For large vocabulary speech recognition, we need to decompose a word into a subword sequence (often called pronunciation modeling), since there are a large number of words. Thus,  $P(\mathbf{X}|\mathbf{W})$  is closely related to phonetic modeling.  $P(\mathbf{X}|\mathbf{W})$  should take into account speaker variations, pronunciation variations, environmental variations, and context-dependent phonetic coarticulation variations. Last, but not least, any static acoustic or language model will not meet the needs of real applications. So it is vital to dynamically adapt both  $P(\mathbf{W})$  and  $P(\mathbf{X}|\mathbf{W})$  to maximize  $P(\mathbf{W}|\mathbf{X})$  while using the spoken language systems. The decoding process of finding the best-matched word sequence,  $\mathbf{W}$ , to match the input speech signal,  $\mathbf{X}$ , in speech-recognition systems is more than a simple pattern recognition problem, since one faces a practically infinite number of word patterns to search in continuous speech recognition.

In the remainder of this section, we will provide an overview on both of  $P(\mathbf{W})$  and  $P(\mathbf{X}|\mathbf{W})$  components in a speech recognizer, as well as on how the maximization operation in Equation 15.2, a process known as decoding, can be carried out in practice.

### 15.2.1 Acoustic Models

The accuracy of automatic speech recognition remains one of the most important research challenges after years of research and development. There are a number of well-known factors that determine the accuracy of a speech-recognition system. The most noticeable ones are context variations, speaker variations, and environment variations. Acoustic modeling plays a critical role to improve the accuracy. It is not far-fetched to state that it is the central part of any speech-recognition system.

AQ2 Acoustic modeling of speech typically refers to the process of establishing statistical representations for the feature vector sequences computed from the speech waveform. HMM (Baum, 1972; Baker, 1975; Jelinek, 1976) is one of the most common types of acoustic models. Other acoustic models include segmental models (Poritz, 1988; Deng, 1993; Deng et al., 1994; Ostendorf et al., 1996; Glass, 2003), super-segmental models including hidden dynamic models (Deng et al., 2006), neural networks (Lippman, 1987; Morgan et al., 2005), maximum entropy models (Gao and Kuo, 2006), and (hidden) conditional random fields (Gunawardana et al., 2006).

AQ3 Acoustic modeling also encompasses “pronunciation modeling,” which describes how a sequence or multi-sequences of fundamental speech units (such as phones or phonetic feature) are used to represent larger speech units such as words or phrases that are the object of speech recognition. Acoustic modeling may also include the use of feedback information from the recognizer to reshape the feature vectors of speech in achieving noise robustness in speech recognition.

In speech recognition, statistical properties of sound events are described by the acoustic model. Correspondingly, the likelihood score  $p(X|W)$  in Equation 15.2 is computed based on the acoustic model. In an isolated-word speech-recognition system that has an  $N$ -word vocabulary, assuming that the acoustic model component corresponding to the  $i$ th word  $W_i$  is  $\lambda_i$ , then  $p(X|W_i) = p(X|\lambda_i)$ . In HMM-based speech recognition, it is assumed that the sequence of observed vectors corresponding to each word is generated by a Markov chain. As shown in Figure 15.3, an HMM is a finite state machine that changes state once every time frame, and at each time frame  $t$  when a state  $j$  is entered, an observation vector  $x_t$  is generated from the emitting probability distribution  $b_j(x_t)$ . The transition property from state  $i$  to state  $j$  is specified by the transition probability  $a_{ij}$ . Moreover, two special non-emitting states are usually used in an HMM. They include an entry state, which is reached before the speech vector generation process begins, and an exit state, which is reached when the generative process terminates. Both states are reached only once. Since they do not generate any observation, none of them has an emitting probability density.

In the HMM, the transition probability  $a_{ij}$  is the probability of entering state  $j$  given the previous state  $i$ , that is,  $a_{ij} = \Pr(s(t) = j | s(t-1) = i)$ , where  $s(t)$  is the state index at time  $t$ . For an  $N$ -state HMM, we have,

$$\sum_{j=1}^N a_{ij} = 1.$$

The emitting probability density  $b_j(x)$  describes the distribution of the observation vectors at the state  $j$ . In continuous-density HMM (CDHMM), emitting probability density is often represented by a Gaussian mixture density:

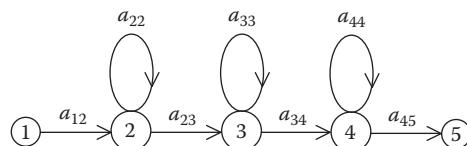
$$b_j(x) = \sum_{m=1}^M c_{jm} N(x; \mu_{jm}, \Sigma_{jm}),$$

where

$$N(x; \mu_{jm}, \Sigma_{jm}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_{jm}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x - \mu_{jm})^T \Sigma_{jm}^{-1} (x - \mu_{jm})}$$

$\Sigma_{jm}$  is a multivariate Gaussian density

$D$  is the dimension of the feature vector  $x$



**FIGURE 15.3** Illustration of a five-state left-to-right HMM. It has two non-emitting states and three emitting states. For each emitting state, the HMM is only allowed to remain at the same state or move to the next state.

$c_{jm}$ ,  $\mu_{jm}$ , and  $\Sigma_{jm}$  are the weight, mean, and covariance of the  $m$ th Gaussian component of the mixture distribution at state  $j$

Generally speaking, each emitting distribution characterizes a sound event, and the distribution must be specific enough to allow discrimination between different sounds as well as robust enough to account for the variability in natural speech.

Numerous HMM training methods are developed to estimate values of the state transition probabilities and the parameters of the emitting probability densities at each state of the HMM. In early years of HMM applications to speech recognition, the EM algorithm, based on the maximum-likelihood principle, as developed in Baum (1972); Baker (1975); Jelinek (1976); Dempster et al. (1977) was typically used as the training method from the training data. The high efficiency of the EM algorithm is one crucial advantage associated with using the HMM as the acoustic model for speech recognition. The effectiveness of the EM for training HMMs was questioned in later research, resulting in a series of more effective but less efficient training algorithms, known as discriminative learning (Bahl et al., 1986; Juang et al., 1997; Macherey et al., 2005; Povey et al., 2005). A comprehensive and unifying review of discriminative learning methods for speech recognition can be found in He et al. (2008).

Given  $\{a_{ij}\}$  and  $b_j(x)$ , for  $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, N$ , the likelihood of an observation sequence  $X$  is calculated as:

$$p(X|\lambda) = \sum_s p(X, s|\lambda), \quad (15.3)$$

where  $s = s_1, s_2, \dots, s_T$  is the HMM state sequence that generates the observation vector sequence  $X = x_1, x_2, \dots, x_T$ , and the joint probability of  $X$  and the state sequence  $s$  given  $\lambda$  is a product of the transition probabilities and the emitting probabilities

$$p(X, s|\lambda) = \prod_{t=1}^T b_{s_t}(x_t) a_{s_t s_{t+1}},$$

where  $s_{T+1}$  is the non-emitting exit state.

In practice, Equation 15.3 can be approximately calculated as joint probability of the observation vector sequence  $X$  with the most possible state sequence, that is,

$$p(X|\lambda) \approx \max_s p(X, s|\lambda). \quad (15.4)$$

Although it is impractical to evaluate the quantities of Equations 15.3 and 15.4 directly due to the huge number of possible state sequences when  $T$  is large, efficient recursive algorithms exist for computing both of them. This is another crucial computational advantage, developed originally in Baum (1972); Baker (1975); Jelinek (1976), for HMMs as an acoustic model for speech recognition.

### 15.2.2 Language Models

The role of language modeling in speech recognition is to provide the value  $P(W)$  in the fundamental equation of speech recognition of Equation 15.2. One type of language model is the grammar, which is a formal specification of the permissible structures for the language. The traditional, deterministic grammar gives the probability of one if the structure is permissible or of zero otherwise. The parsing technique, as discussed in Chapter 4, is the method of analyzing the sentence to see if its structure is compliant with the grammar. With the advent of bodies of text (corpora) that have had their structures hand-annotated, it is now possible to generalize the formal grammar to include accurate probabilities. Furthermore, the probabilistic relationship among a sequence of words can be directly derived and modeled from the

AQ4

AQ5

corpora with the so-called stochastic language models, such as  $n$ -gram, avoiding the need to create broad coverage formal grammars.

Another, more common type of language model is called the stochastic language model, which plays a critical role in building a working spoken language system. We will discuss a number of important issues associated with this type of language models.

As covered earlier, a language model can be formulated as a probability distribution  $P(\mathbf{W})$  over word strings  $\mathbf{W}$  that reflect how frequently a string  $\mathbf{W}$  occurs as a sentence. For example, for a language model describing spoken language, we might have  $P(hi) = 0.01$  since perhaps one out of every hundred sentences a person speaks is *hi*. On the other hand, we would have  $P(lid gallops Changsha pop) = 0$  since it is extremely unlikely anyone would utter such a strange string.

$P(\mathbf{W})$  can be decomposed as

$$\begin{aligned} P(\mathbf{W}) &= P(w_1, w_2, \dots, w_n) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \cdots P(w_n|w_1, w_2, \dots, w_{n-1}) \\ &= \prod_{i=1}^n P(w_i|w_1, w_2, \dots, w_{i-1}) \end{aligned} \quad (15.5)$$

where  $P(w_i|w_1, w_2, \dots, w_{i-1})$  is the probability that  $w_i$  will follow given that the word sequence  $w_1, w_2, \dots, w_{i-1}$  was presented previously. In Equation 15.5, the choice of  $w_i$  thus depends on the entire past history of the input. For a vocabulary of size  $v$  there are  $v^{i-1}$  different histories and so, to specify  $P(w_i|w_1, w_2, \dots, w_{i-1})$  completely,  $v^i$  values would have to be estimated. In reality, the probabilities  $P(w_i|w_1, w_2, \dots, w_{i-1})$  are impossible to estimate for even moderate values of  $i$ , since most histories  $w_1, w_2, \dots, w_{i-1}$  are unique or have occurred only a few times. A practical solution to the above problems is to assume that  $P(w_i|w_1, w_2, \dots, w_{i-1})$  only depends on some equivalence classes. The equivalence class can be simply based on the several previous words  $w_{i-N+1}, w_{i-N+2}, \dots, w_{i-1}$ . This leads to an *N-gram* language model. If the word depends on the previous two words, we have a *trigram*:  $P(w_i|w_{i-2}, w_{i-1})$ . Similarly, we can have *unigram*:  $P(w_i)$ , or *bigram*:  $P(w_i|w_{i-1})$  language models. The trigram is particularly powerful as most words have a strong dependence on the previous two words and it can be estimated reasonably well with an attainable corpus.

In bigram models, we make the approximation that the probability of a word depends only on the identity of the immediately preceding word. To make  $P(w_i|w_{i-1})$  meaningful for  $i = 1$ , we pad the *beginning of the sentence* with a distinguished token  $\langle s \rangle$ ; that is, we pretend  $w_0 = \langle s \rangle$ . In addition, to make the sum of the probabilities of all strings equal 1, it is necessary to place a distinguished token  $\langle /s \rangle$  at the *end of the sentence*. For example, to calculate  $P(Mary loves that person)$  we would take

$$\begin{aligned} P(Mary loves that person) &= \\ P(Mary|\langle s \rangle)P(loves|Mary)P(that|loves)P(person|that)P(\langle /s \rangle|person) \end{aligned}$$

To estimate  $P(w_i|w_{i-1})$ , the frequency with which the word  $w_i$  occurs given that the last word is  $w_{i-1}$ , we simply count how often the sequence  $P(w_i|w_{i-1})$  occurs in some text and normalize the count by the number of times  $w_{i-1}$  occurs.

In general, for a trigram model, the probability of a word depends on the two preceding words. The trigram can be estimated by observing the frequencies or counts of the word pair  $C(w_{i-2}, w_{i-1})$  and triplet  $C(w_{i-2}, w_{i-1}, w_i)$  as follows:

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})} \quad (15.6)$$

The text available for building a model is called a training corpus. For  $n$ -gram models, the amount of training data used is typically many millions of words. The estimate of Equation 15.6 is based on the

maximum likelihood principle, because this assignment of probabilities yields the trigram model that assigns the highest probability to the training data of all possible trigram models.

We sometimes refer to the value  $n$  of an  $n$ -gram model as its order. This terminology comes from the area of Markov models, of which  $n$ -gram models are an instance. In particular, an  $n$ -gram model can be interpreted as a Markov model of order  $n - 1$ .

Consider a small example. Let our training data  $S$  be comprised of the three sentences *John read her book. I read a different book. John read a book by Mulan* and let us calculate  $P(\text{John read a book})$  for the maximum likelihood bigram model. We have

$$\begin{aligned} P(\text{John}|<\text{s}>) &= \frac{C(<\text{s}>, \text{John})}{C(<\text{s}>)} = \frac{2}{3} \\ P(\text{read}|\text{John}) &= \frac{C(\text{John}, \text{read})}{C(\text{John})} = \frac{2}{2} \\ P(\text{a}|\text{read}) &= \frac{C(\text{read}, \text{a})}{C(\text{read})} = \frac{2}{3} \\ P(\text{book}|\text{a}) &= \frac{C(\text{a}, \text{book})}{C(\text{a})} = \frac{1}{2} \\ P(</\text{s}>|\text{book}) &= \frac{C(\text{book}, </\text{s}>)}{C(\text{book})} = \frac{2}{3} \end{aligned}$$

These trigram probabilities help us to estimate the probability for the sentence as:

$$\begin{aligned} P(\text{John}, \text{read}, \text{a}, \text{book}) &= P(\text{John}|<\text{s}>)P(\text{read}|\text{John})P(\text{a}|\text{read})P(\text{book}|\text{a})P(</\text{s}>|\text{book}) \\ &\approx 0.148 \end{aligned} \quad (15.7)$$

If these three sentences are all the data we have to train our language model, the model is unlikely to generalize well to new sentences. For example, the probability for *Mulan read her book* should have a reasonable probability, but the trigram will give it a zero probability simply because we do not have a reliable estimate for  $P(\text{read}|\text{Mulan})$ .

Unlike linguistics, grammaticality is not a strong constraint in the  $n$ -gram language model. Even though the string is ungrammatical, we may still assign it a high probability if  $n$  is small.

Language can be thought of as an information source whose outputs are words  $w_i$  belonging to the vocabulary of the language. The most common metric for evaluating a language model is the word recognition error rate, which requires the participation of a speech-recognition system. Alternatively, we can measure the probability that the language model assigns to test word strings without involving speech-recognition systems. This is the derivative measure of cross-entropy known as test set *perplexity*.

Given a language model that assigns probability  $P(\mathbf{W})$  to a word sequence  $\mathbf{W}$ , we can derive a compression algorithm that encodes the text  $\mathbf{W}$  using  $-\log_2 P(\mathbf{W})$  bits. The cross-entropy  $H(\mathbf{W})$  of a model  $P(w_i|w_{i-n+1} \dots w_{i-1})$  on data  $\mathbf{W}$ , with a sufficiently long word sequence, can be simply approximated as

$$H(\mathbf{W}) = -\frac{1}{N_{\mathbf{W}}} \log_2 P(\mathbf{W}) \quad (15.8)$$

where  $N_{\mathbf{W}}$  is the length of the text  $\mathbf{W}$  measured in words.

The perplexity  $PP(\mathbf{W})$  of a language model  $P(\mathbf{W})$  is defined as the reciprocal of the (geometric) average probability assigned by the model to each word in the test set  $\mathbf{W}$ . This is a measure, related to cross-entropy, known as test-set perplexity:

$$PP(\mathbf{W}) = 2^{H(\mathbf{W})} \quad (15.9)$$

The perplexity can be roughly interpreted as the geometric mean of the branching factor of the text when presented to the language model. The perplexity defined in Equation 15.9 has two key parameters: a language model and a word sequence. The test-set perplexity evaluates the generalization capability of the language model. The training-set perplexity measures how the language model fits the training data, such as the likelihood. It is generally true that lower perplexity correlates with better recognition performance. This is because the perplexity is essentially a statistically weighted word branching measure on the test set. The higher the perplexity, the more branches the speech recognizer needs to consider statistically.

While the perplexity defined in Equation 15.9 is easy to calculate for the  $n$ -gram (Equation 15.5), it is slightly more complicated to compute it for a probabilistic CFG. We can first parse the word sequence and use Equation 15.5 to compute  $P(W)$  for the test set perplexity. The perplexity can also be applied to non-stochastic models such as CFGs. We can assume they have a uniform distribution in computing  $P(W)$ .

A language with higher perplexity means the number of words branching from a previous word is larger on average. In this sense, the perplexity is an indication of the complexity of the language if we have an accurate estimate of  $P(W)$ . For a given language, the difference between the perplexity of a language model and the true perplexity of the language is an indication of the quality of the model. The perplexity of a particular language model can change dramatically in terms of the vocabulary size, the number of states or grammar rules, and the estimated probabilities. A language model with perplexity  $X$  has roughly the same difficulty as another language model in which every word can be followed by  $X$  different words with equal probabilities. Therefore, in the task of continuous digit recognition, the perplexity is 10. Clearly, lower perplexity will generally have less confusion in recognition. Typical perplexities yielded by  $n$ -gram models on English text range from about 50 to almost 1000 (corresponding to cross-entropies from about 6 to 10 bits/word), depending on the type of text. In tasks of 5000 word continuous speech recognition for the *Wall Street Journal* newspaper, the test set perplexity of the trigram grammar and the bigram grammar is reported to be about 128 and 176 respectively. In tasks of 2000 word conversational Air Travel Information System (ATIS), the test set perplexity of the word trigram model is typically less than 20.

Since perplexity does not take into account the acoustic confusability, we eventually have to measure speech-recognition accuracy. For example, if the vocabulary of a speech recognizer contains the E-set of English alphabet:  $B, C, D, E, G$ , and  $T$ , we can define a CFG that has a low perplexity value of 6. Such a low perplexity does not guarantee we will have good recognition performance, because of the intrinsic acoustic confusability of the E-set.

### 15.2.3 Decoding

As epitomized in the fundamental equation of speech recognition in Equation 15.2, the decoding process in a speech recognizer's operation is to find a sequence of words whose corresponding acoustic and language models best match the input feature vector sequence. Therefore, the process of such a decoding process with trained acoustic and language models is often referred to as a *search* process. Graph search algorithms have been explored extensively in the fields of artificial intelligence, operating research, and game theory, which serve as the basic foundation for the search problem in continuous speech recognition.

The complexity of a search algorithm is highly correlated to the search space, which is determined by the constraints imposed by the language models. The impact of different language models, including finite-state grammars, CFG, and  $n$ -grams are critical to decoding efficiency.

Speech recognition search is usually done with the Viterbi decoder (Viterbi, 1967; Vintsyuk, 1968; Sakoe and Chiba, 1971; Ney, 1984), or A\* stack decoder (Jelinek, 1969, 1976, 1997). The reasons for choosing the Viterbi decoder involve arguments that point to speech as a left to right process and the efficiencies afforded by a time-synchronous process. The reasons for choosing a stack decoder involve its ability to more effectively exploit the A\* criteria that holds out the hope of performing an optimal search as well as the ability to handle huge search spaces. Both algorithms have been successfully applied to various speech-recognition systems. Viterbi beam search has been the preferred method for almost all

speech recognition tasks. Stack decoding, on the other hand, remains an important strategy to uncover the n-best and lattice structures (Schwartz and Chow, 1990).

The decoder uncovers the word sequence  $\hat{\mathbf{W}} = w_1 w_2 \dots w_m$  that has the maximum posterior probability  $P(\mathbf{W}|\mathbf{X})$  for the given acoustic observation  $\mathbf{X} = X_1 X_2 \dots X_n$ , according to the maximization operation described by Equation 15.2. One obvious (and brute-force) way is to search all possible word sequences and select the one with best posterior probability score. This, however, is not practically feasible.

The unit of acoustic model  $P(\mathbf{X}|\mathbf{W})$  is not necessarily a word model. For large-vocabulary speech-recognition systems, sub-word models, which include phonemes, demi-syllables, and syllables are often used. When sub-word models are used, the word model  $P(\mathbf{X}|\mathbf{W})$  is then obtained by concatenating the sub-word models according to the pronunciation transcription of the words in a lexicon or dictionary.

When word models are available, speech recognition becomes a search problem. The goal for speech recognition is thus to find a sequence of word models that best describes the input waveform against the word models. As neither the number of words nor the boundary of each word or phoneme in the input waveform is known, appropriate search strategies to deal with these variable-length nonstationary patterns are extremely important.

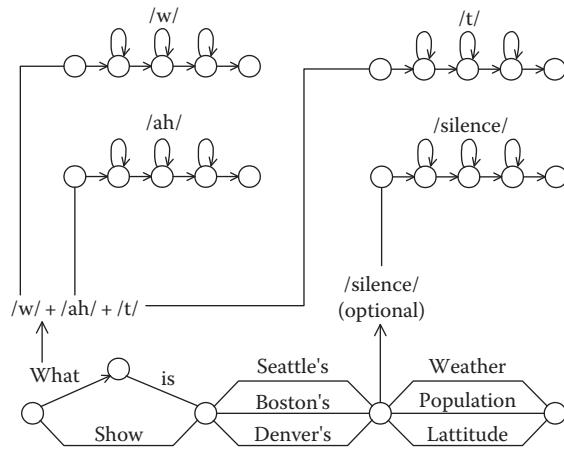
When HMMs are used for speech-recognition systems, the states in the HMM can be expanded to form the state-search space in the search. Here, we use HMM as our speech models. Although the HMM framework is used to describe the search algorithms, the techniques discussed here can in principle be used for systems based on other modeling techniques. The HMM's state transition network is sufficiently general and it can represent the general search framework for most modeling approaches.

### 15.2.3.1 The Concept of State Space in Decoding

The state space in speech recognition search or decoding is an important concept, as it is a good indicator of the complexity for the search. Since the HMM representation for each word in the lexicon is fixed, the state space and its size are determined by the language models. And each language model (grammar) can be correlated to a state machine that can be expanded to form the full state space for the recognizer. The states in such a state machine are referred to as *language models states*. For simplification, we will use the concepts of state space and language model states interchangeably. The expansion of language model states to HMM states will be done implicitly. The language model states for isolated word recognition are trivial. They are just the union of the HMM states of each word. In this section, we first look at the language model states for two grammars for continuous speech recognition: Finite State Grammar (FSG) and Context Free Grammar (CFG). We then discuss a most popular decoding technique, time-synchronous beam search technique.

In general, the decoding complexity for the time-synchronous Viterbi algorithm is  $O(N^2 T)$  where  $N$  is the total number states in the composite HMM and  $T$  is the length of the input observation. A full time-synchronous Viterbi search is quite efficient for moderate tasks (vocabulary  $\leq 500$ ). Most small or moderate vocabulary tasks in speech-recognition applications use an FSG. Figure 15.4 shows a simple example of a FSG, where each of the word arcs in a FSG can be expanded as a network of phoneme or other sub-word HMMs. The word HMMs are connected with null transitions with the grammar state. A large finite state HMM network that encodes all the legal sentences can be constructed based on the expansion procedure. The decoding process is achieved by performing the time-synchronous Viterbi search on this composite finite state HMM.

In practice, FSGs are sufficient only for simple tasks. When a FSG is made to satisfy the constraints of sharing of different sub-grammars for compactness and support for dynamic modifications, the resulting nondeterministic FSG becomes similar to CFG in terms of implementation. The CFG grammar consists of a set of productions or rules, which expand nonterminals into a sequence of terminals and nonterminals. Nonterminals in the grammar tend to refer to high-level task-specific concepts such as dates, names, and commands. The terminals are words in the vocabulary. A grammar also has a nonterminal designated as its start state. While CFG has not been widely used in the NLP community, they are one of the most widely used methods for speech recognition.



**FIGURE 15.4** An illustration of how to compile a speech-recognition task with finite grammar into a composite HMM.

A CFG can be formulated with a recursive transition network (RTN). RTN allows arc labels to refer to other networks as well as words. We use Figure 15.5 to illustrate how to embed HMMs into a RTN, which represents the following CFG:

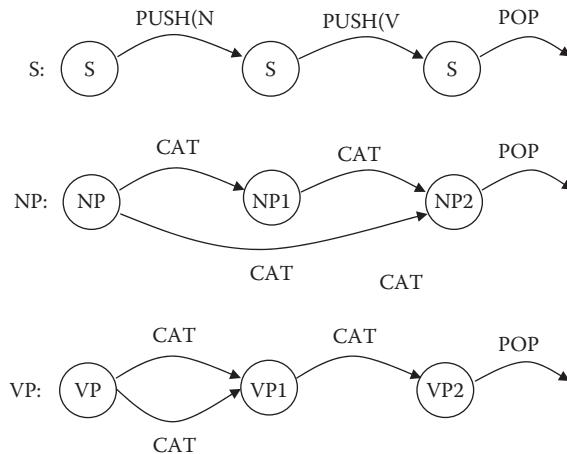
```

S → NP VP
NP → sam|sam davis
VP → VERB tom
VERB → likes|hates
    
```

There are three types of arcs in a RTN shown in Figure 15.5: CAT( $x$ ), PUSH( $x$ ), POP.CAT( $x$ ) arc indicates  $x$  is a terminal node (which is equivalent to a word arc). Therefore, all the CAT( $x$ ) arcs can be expanded by the HMM network for  $x$ . The word HMM can again be composite HMM built from phoneme (or subword) HMMs. Similar to the finite state grammar case in Figure 15.4, all grammar states act as a state with incoming and outgoing null transitions to connect word HMMs in the CFG. During decoding, the search pursues several paths through the CFG at the same time. Associated with each of the paths is a grammar state that describes completely how the path can be extended further. When the decoder hypothesizes end of the current word of a path, it asks the CFG module to extend the path further by one word. There may be several alternative successor words for the given path. The decoder considers all the successor word possibilities. This may cause the path to be extended to generate several more paths to be considered each with its own grammar state.

Readers should note that the same word might be under consideration by the decoder in the context of different paths and grammar states at the same time. For example, there are two word arcs CAT (Sam) in Figure 15.5. Their HMM states should be considered as distinct states in the trellis because they are in completely different grammar states. Two different states in trellis also mean different paths going into these two states cannot be merged. Since these two partial paths will lead to different successive paths, the search decision needs to be postponed until the end of search. Therefore, when embedding HMM into word arc in grammar network, the HMM state will be assigned a new state identity although the HMM parameters (transition probabilities and output distributions) can still be shared across different grammar arcs.

Each path consists of a stack of production rules. Each element of the stack also contains the position within the production rule of the symbol that is currently being explored. The search graph (trellis) started from the initial state of CFG (state S). When the path needs to be extended, we look at the next



**FIGURE 15.5** An simple RTN example with three types of arcs:  $\text{CAT}(x)$ ,  $\text{PUSH}(x)$ , and  $\text{POP}$ .

arc (symbol in CFG) in the production. When the search enters a  $\text{CAT}(x)$  arc (terminal), the path gets extended with the terminal and the HMM trellis computation is performed on the  $\text{CAT}(x)$  arc to match the model  $x$  against the acoustic data. When the final state of the HMM for  $x$  is reached, the search moves on via the null transition to the destination of the  $\text{CAT}(x)$  arc. When the search enters a  $\text{PUSH}(x)$  arc, it indicates a nonterminal symbol  $x$  is encountered. In effect, the search is about to enter a subnetwork of  $x$ , the destination of the  $\text{PUSH}(x)$  arc is stored in a last in first out (LIFO) stack. When the search reaches a  $\text{POP}$  arc, the search returns to the state extracted from the top of the LIFO stack. Finally, when we reach the end of the production rule at the very bottom of the stack, we have reached an accepting state in which we have seen a complete grammatical sentence. For our decoding purpose, that is the state we want to pick as the best score at the end time frame  $T$  to obtain the search result.

The problem of connected word recognition by FSG or CFG is that the number of states increases enormously when it is applied to complex tasks and grammars. Moreover, it remains a challenge to generate such a FSG or a CFG from a large corpus, either manually or automatically. Finally, it is questionable if FSG or CFG is adequate to describe natural languages or unconstrained spontaneous languages. Instead,  $n$ -gram language models, which we described earlier in this section, are often used for natural languages or unconstrained spontaneous languages.

### 15.2.3.2 Time-Synchronous Viterbi Search

When HMMs are used for acoustic models, the acoustic model score (likelihood) used in search is by definition the forward probability. That is, all possible state sequences must be considered. Thus,

$$P(\mathbf{X}|\mathbf{W}) = \sum_{\text{all possible } s_0^T} P(\mathbf{X}, s_0^T | \mathbf{W})$$

where the summation is to be taken over all possible state sequences  $\mathbf{S}$  with the word sequence  $\mathbf{W}$  under consideration. However, under the trellis framework, more bookkeeping must be performed since we cannot add scores with different word sequence history. Since the goal of decoding is to uncover the best word sequence, we could approximate the summation with the maximum to find the best state sequence instead. We then have the following approximation:

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{w}} P(\mathbf{W}) P(\mathbf{X}|\mathbf{W}) \cong \arg \max_{\mathbf{w}} \left\{ P(\mathbf{W}) \max_{s_0^T} P(\mathbf{X}, s_0^T | \mathbf{W}) \right\}$$

which is often referred to as the *Viterbi approximation*. It can literally translated to “the *most likely word sequence* is approximated by the *most likely state sequence*.” The Viterbi search is then suboptimal. Although the search results by using the forward probability and the Viterbi probability could in principle be different, in practice this is very rare to be the case.

The Viterbi search can be executed very efficiently via the trellis framework. It is a time-synchronous search algorithm that completely processes time  $t$  before going on to time  $t + 1$ . For time  $t$ , each state is updated by the best score (instead of the sum of all incoming paths) from all states in at time  $t - 1$ . This is why it is often called the *time-synchronous Viterbi search*. When one update occurs, it also records the backtracking pointer to remember the most probable incoming state. At the end of the search, the most probable state sequence can be recovered by tracing back these backtracking pointers. The Viterbi algorithm provides an optimal solution for handling nonlinear time warping between HMMs and the acoustic observation, word boundary detection, and word identification in continuous speech recognition. This unified Viterbi search algorithm serves as the fundamental technique for most search algorithms in use in continuous speech recognition.

It is necessary to clarify the backtracking pointer for the time-synchronous Viterbi search for continuous word recognition. Actually, we are not interested in the optimal state sequence per se. Instead, we are only interested in the optimal word sequence. Therefore, we use the backtrack pointer to just remember the word history for the current path, so the optimal word sequence can be recovered at the end of the search. To be more specific, when we reach the final state of a word, we create a history node containing the word identity and current time index and append this history node to the existing backtrack pointer. This backtrack pointer is then passed onto the successor node if it is the optimal path leading to the successor node for both intra-word and inter-word transition. The side benefit of keeping this backtrack pointer is that we no longer need to keep the entire trellis during the search. Instead, we only need space to keep two time-slices (columns) in the trellis computation (the previous time slice and the current time slice) because all the backtracking information is now kept in the backtrack pointer. This simplification is a significant benefit in the implementation of a time-synchronous Viterbi search.

The time-synchronous Viterbi search can be considered as a *breadth first search* with dynamic programming. Instead of performing a tree search algorithm, the dynamic programming principle helps to create a search graph where multiple paths leading to the same search state are merged by keeping the best path (with the minimum cost). The Viterbi trellis is a representation of the search graph. Therefore, all efficient techniques for graph search algorithms can be applied to the time-synchronous Viterbi search. Although we have so far described the trellis in an explicit fashion, where the entire search space needs to be explored before the optimal path can be found, it is not necessary to do so. When the search space contains an enormous number of states, it becomes impractical to pre-compile the composite HMM entirely and store it in the memory. It is preferable to dynamically build and allocate portions of the search space that is sufficient to search the promising paths. By using the graph search algorithm, only part of the entire Viterbi trellis is generated explicitly. By constructing the search space dynamically, the computation cost of the search is proportional only to the number of active hypotheses that is independent of the overall size of the potential search space. Therefore, dynamically generated trellis is a key to the heuristic Viterbi search for efficient large-vocabulary continuous speech recognition.

### 15.3 Major Historical Developments in Speech Recognition

Each of the above three components in speech-recognition technology has experienced significant historical development. In fact, the establishment of the basic statistical framework, as epitomized by the fundamental equation of speech recognition described in the preceding section in Equation 15.1 or 15.2, constitutes one major milestone in the historical development of speech recognition. In the following, we review and highlight this and other developments in the field, based partly on the recently published materials in Baker et al. (2007, 2009a,b).

In the first category of the significant historical developments in speech recognition are the establishment of the statistical paradigm, and the associated models and algorithms enabling the implementation of the paradigm. The most significant paradigm shift for speech-recognition progress has been the change from the earlier nonstatistical methods to statistical ones, especially stochastic processing with HMMs (Baker, 1975; Jelinek, 1976) introduced as an acoustic modeling component of speech recognition in the early 1970s. More than 30 years later, this methodology still remains as the predominant one. A number of models and algorithms have been efficiently incorporated within this framework. The Expectation-Maximization (EM) Algorithm and the Forward-Backward or the Baum-Welch algorithm (Baum, 1972; Dempster et al., 1977) have been the basic and principal means by which the HMMs are trained highly efficiently from data. Similarly, for the language-modeling component,  $N$ -gram language models and the variants, trained with the basic counting or EM-style techniques, have proved remarkably powerful and resilient. Beyond these basic algorithms, statistical discriminative training techniques have been developed since late 1980s based not on the likelihood for data-matching criteria but on maximum mutual information or related minimum error criteria (Bahl et al., 1986; Povey et al., 2005; He et al., 2008). And beyond the basic HMM-like acoustic models and basic  $N$ -gram-like language models, further developments include segmental models (Poritz, 1988; Deng et al., 1993, 1994; Ostendorf et al., 1996; Glass, 2003), and structured speech and language models (Chelba and Jelinek, 2000; Wang et al., 2000; Deng et al., 2006). Despite continuing work in this area, however, large-scale success is yet to be demonstrated.

Another important area of algorithm development is adaptation, which is vital to accommodating a wide range of variable conditions for the channel, noise, speaker, vocabulary, accent, and recognition domain, etc. Effective adaptation algorithms enable rapid application integration, and are a key to the successful commercial deployment of speech-recognition technology. The most popular adaptation techniques include Maximum a Posteriori probability (MAP) estimation (Gauvain and Lee, 1994) and Maximum Likelihood Linear Regression (MLLR) (Leggetter and Woodland, 1995). Training can take place on the basis of small amounts of data from new tasks or domains for additional training material, as well as “one-shot” learning or “unsupervised” learning at test time (Huang and Lee, 1993). These adaptation techniques have also been generalized to train the “generic” models so that they are better able to represent the overall statistics of the full training data set, a technique called Speaker Adaptive Training or SAT (Anastasakos et al., 1997).

AQ6

AQ7

In the second category of the significant advancement in speech recognition is the establishment of the computational infrastructure that enables the above statistical model/algorithm developments. Moore’s Law observes long-term progress in computer development, and predicts doubling the amount of computation for a given cost every 12–18 months, as well as a comparably shrinking cost of memory. These have been instrumental in enabling speech-recognition researchers to develop and evaluate complex algorithms on sufficiently large tasks in order to make realistic progress. In addition, the availability of common speech corpora for speech training, development, and evaluation, has been critical, allowing the creation of complex systems of ever-increasing capabilities. Since speech is a highly variable signal and is characterized by many parameters, large corpora become critical in modeling it well enough for automated systems to achieve proficiency. Over the years, these corpora have been created, annotated, and distributed to the worldwide community by the National Institute of Standard and Technology (NIST), the Linguistic Data Consortium (LDC), European Language Resources Association (ELRA), and other organizations. The character of the recorded speech has progressed from limited, constrained speech materials to huge amounts of progressively more realistic, spontaneous speech. The development and adoption of rigorous benchmark evaluations and standards, nurtured by NIST and others, have also been critical in developing increasingly powerful and capable systems. Many labs and researchers have benefited from the availability of common research tools such as HTK, Sphinx, CMU LM toolkit, and SRILM toolkit. Extensive research support combined with workshops, task definitions, and system evaluations sponsored by DARPA (the U.S. Department of Defense Advanced Research Projects Agency) and others have been essential to today’s system developments.

Historically significant advancement of speech recognition has the third category that we call knowledge representation. This includes the development of perceptually motivated speech signal representations such as Mel-Frequency Cepstral Coefficients (MFCC) (Davis and Mermelstein, 1980) and Perceptual Linear Prediction (PLP) coefficients (Hermansky, 1990), as well as normalizations via Cepstral Mean Subtraction (CMS) (Rosenberg et al., 1994), RASTA (Hermansky and Morgan, 1994), and Vocal Tract Length Normalization (VTLN) (Eide and Gish, 1996). Architecturally, the most important development in knowledge representation has been searchable unified graph representations that allow multiple sources of knowledge to be incorporated into a common probabilistic framework. Non-compositional methods include multiple speech streams, multiple probability estimators, multiple recognition systems combined at the hypothesis level, for example, ROVER (Fiscus, 1997), and multi-pass systems with increasing constraints (bigram vs. four-gram, within word dependencies vs. cross-word, etc). More recently, the use of multiple algorithms, applied both in parallel and sequentially has proven fruitful, as have multiple types of feature-based transformations such as heteroscedastic linear discriminant analysis (HLDA) (Kumar and Andreou, 1998), feature-space minimum phone error (fMPE) (Povey et al., 2005), and neural net-based features (Morgan et al., 2005).

The final category of major historical significant developments in speech recognition includes key decoding or search strategies that we have discussed earlier in this section. These strategies have focused on the stack decoding ( $A^*$  search) (Jelinek, 1969) and the time-synchronous Viterbi search (Viterbi, 1967; Vintsyuk, 1968; Sakoe and Chiba, 1971; Ney, 1984). Without these practical decoding algorithms, large-scale continuous speech recognition would not be possible.

## 15.4 Speech-Recognition Applications

---

The ultimate impact of speech recognition depends on whether one can fully integrate the enabling technologies with applications. How to effectively integrate speech into applications often depends on the nature of the user interface and application. In discussing some general principles and guidelines in developing spoken language applications, we must look closely at designing the user interface.

A well-designed user interface entails carefully considering the particular user group of the application and delivering an application that works effectively and efficiently. As a general guideline, one needs to make sure that the interface matches the way users want to accomplish a task. One also needs to use the most appropriate modality at the appropriate time to assist users to achieve their goals. One unique challenge in speech-recognition applications is that speech recognition (as well as understanding) is imperfect. In addition, the spoken command can be ambiguous so a dialogue strategy is necessary to clarify the goal of the speaker. There are always mistakes one has to deal with. It is critical that applications employ necessary interactive error handling techniques to minimize the impact of these errors. Application developers should therefore fully understand the strengths and weaknesses of the underlying speech technologies and identify the appropriate place to use speech recognition and understanding technology effectively.

There are three broad classes of applications: (1) Cloud-based call center/IVR (Interactive Voice Response): This includes the widely used applications from Tellme's information access over the phone to Microsoft Exchange Unified Messaging; (2) PC-based dictation/command and control: There are a number of dictation applications on the PC. It is a useful tool for accessibility benefits, but not yet ready for the mainstream. (3) Device-based embedded command control: There is a wide range of devices that do not have a typical PC keyboard or mouse, and the traditional GUI application cannot be directly extended. As an example, Microsoft's Response Point blue button illustrates what speech interface can do to make the user interface much simpler. Mobile phones and automobile scenarios are also very suitable for speech applications. Because of the physical size and hands-free and eyes-free constraints, the traditional GUI application interaction model requires a significant modification. Ford Sync is a



**FIGURE 15.6** Ford SYNC highlights the car's speech interface—“You Talk. SYNC listens.”



**FIGURE 15.7** Windows Live Local Search highlights speech functions—Just say what you’re looking for!

good example on leveraging the power of speech technologies in this category (Figure 15.6). Voice search is also highly suitable for mobile phones (Figure 15.7).

AQ8

Speech interface has the potential to provide a consistent and unified interaction model across these scenarios. The increased mobility of our society demanded access to information and services at anytime and anywhere. Both cloud and client-based voice-enabled applications can vastly enhance the user experience and optimize cost savings for businesses. We here in this section selectively take some examples as case studies to illustrate real-world applications and challenges.

### 15.4.1 IVR Applications

Given that speech recognizers make mistakes, the inconvenience to the user must be minimized. This means that careful design of human/machine interaction (e.g., the call flow) is essential in providing reliable and acceptable services. Hence, a demand for skilled user interface designers has occurred over the past decade.

Because of the increased adoption of the Web, IVR-based interaction provides a more ubiquitous but less effective information access than Web-based browsing. Because the phone is widely available, there is an important empirical principle to decide if the IVR should be deployed. If the customer can wait for more than two hours to get the information, the need to have the speech-based IVR would be less critical. This is because the Web has been very pervasive and generally provides a better user interface for customers to access the information. If the task is time sensitive, and the customer may be on the move without having the access to a PC, such IVR applications would bridge the gap and provide a complementary value. The key benefit of network-based IVR services is to provide the user with access to information independent of the user's communication device or location.

In 2005, Forrester Research published a study reporting that on average, call center human agents can cost \$5.50–\$12.00 per call and speech-enabled IVR services could reduce that down to \$0.20/call. With significantly improved natural speech applications, businesses can gain increased call completion rates and happier customers. Successful network-based voice-enabled services have been and continue to be those that have the benefit of simplicity. Successful voice-enabled services are natural for phone-based applications and services. They are easy to use and provide real value to the businesses. Many applications are extensions of DTMF-based IVR services. Speech recognition is used as a touchtone replacement for IVR menus. There are also applications specifically designed for speech-based applications. Microsoft's Tellme's service is such an example.

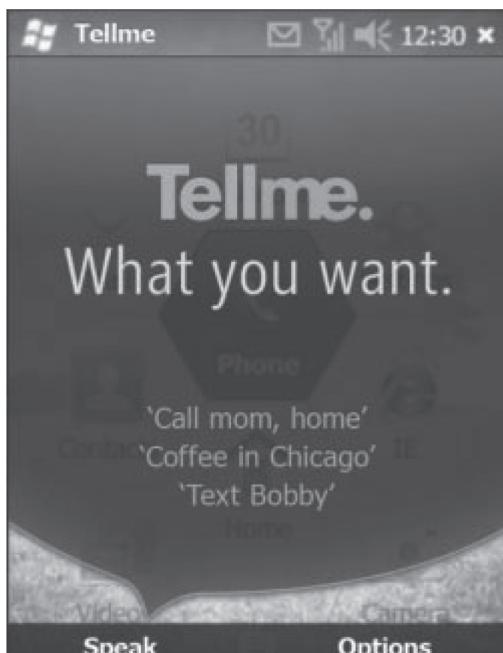
### 15.4.2 Appliance—“Response Point”

With a large of number of diverse devices, proliferating, speech-based user interface becomes increasingly important as we cannot attach a keyboard or mouse to all of these devices. Among these devices, phones offer a unique opportunity for speech recognition as they are designed for voice communications equipped with a well-designed speaker and microphone already. Speech is such a natural way for information exchange on the phone. Early applications such as phone dialing have been available on many embedded devices.

One latest device-based speech recognition example is Microsoft's Response Point phone system designed specifically for small business customers (see Figure 15.8). Response Point is a PBX system that runs on an embedded device without having any moving parts such as hard disk or cooling fan. It provides a comprehensive telephony solution for small business including PBX functions, unified messaging, computer telephony integration and basic IVR for finding people, location, and business hours. The key differentiation feature is the unique blue button on every Response Point phone as shown in Figure 15.9. The blue button allows speakers to simply push the button and issue command for a wide range of communications tasks such as name dialing, transferring the call, and checking voice mails. Response Point is very simple to use, and helps to bring speech applications to the non-technological savvy customers.

### 15.4.3 Mobile Applications

The HMM technology has proven to be an effective method for mobile phones including dictation and name dialing. For example, Nuance offers the HMM-based speaker-independent dialer in a wide range of mobile phones. Client-based speech recognition has the benefit of low-latency.



**FIGURE 15.8** Microsoft's Tellme is integrated into Windows Mobile at the network level.



**FIGURE 15.9** Microsoft's Response Point phone system designed specifically for small business customers.

Ford SYNC is a factory-installed, in-car communications and entertainment system. The system runs on most Ford, Lincoln and Mercury vehicles. Ford SYNC allows drivers to bring nearly any mobile phone or digital media player into their vehicle and operate them using voice commands, the vehicle's steering wheel, or radio controls.

One special type of applications in the mobile environment is voice search (Wang et al., 2008). Voice search provides mobile phone users with the information they request with a spoken query. The information normally exists in a large database, and the query has to be compared with a field in the database to obtain the relevant information. The contents of the field, such as business or product names, are often unstructured text. While general voice search accuracy is not usable, structured voice

search with constrained contexts have been on the market. Google Voice Local Search is now live and publicly available. Microsoft has Windows Live local search that is speech-enabled. Automated voice-based mobile search establishes a clear competitive landscape, which will likely mean a further decline in call volumes and revenues for traditional mobile directory assistance, as consumers become more aware of the availability of these free services.

Windows Live local search is a multimodal application. The widespread availability of broadband access and powerful mobile devices are fostering a new wave of human computer interfaces that support multiple modalities, thus bridging and eventually blurring the device and network voice-enabled service markets. Multimodal interfaces also solve many of today's limitations with speech applications. A picture is worth a thousand words. We believe "speech-in" and "picture-out" takes advantage of the two most natural human modalities that significantly enhance the user experience when dealing with complex applications.

As a conclusion of this section, we will see greater roles of speech recognition in the future's anytime, anywhere, and any-channel communications. The greater automation of telecommunications services, and for individuals, easier access to information and services at any time, with any device, and from anywhere, as well as in any language will be the norm. Customers will benefit from rapid and personalized information access, partly empowered by speech recognition and the related technologies. Speech recognition has just scratched the surface. But to enable this ultimate success, difficult challenges need to be overcome and intensive research is need, which is the subject of the next section.

## 15.5 Technical Challenges and Future Research Directions

---

Despite successful applications of speech recognition in the marketplace and people's lives as described above, the technology is far from being perfect and technical challenges abound. Some years ago (2003–2004), the authors of this chapter have identified two main technical challenges in adopting speech recognition: (1) making speech-recognition systems robust in noisy acoustic environments, and (2) creating workable speech-recognition systems for natural, free-style speech (Deng and Huang, 2004). Since then, huge stride has been made in overcoming these challenges, and yet the problems remain unsolved. In this section, we will address the remaining problems and expand the discussion to include a number of related and new challenges. We also discuss fertile areas for future, longer-term research in speech recognition.

### 15.5.1 Robustness against Acoustic Environments and a Multitude of Other Factors

As discussed in the preceding section, state-of-the-art speech recognition has been built upon a solid statistical framework after a series of successful historical developments. The statistical framework requires probabilistic models, with parameters estimated from sample speech data, which represents the variability that occurs in the natural speech data. These probabilistic models seek to recover linguistic information, such as the words or phrases uttered, from the speech signal received by microphones placed under natural acoustic environments. One key underlying challenge to speech recognition technology is the special complexity of the variability that exists in the natural speech signal. How to identify and handle a multitude of variability factors, some are related to and others are unrelated to the linguistic information being sought by the speech-recognition system, forms one principal source of technical difficulties in building successful speech-recognition systems.

One pervasive type of variability in the speech signal that is typically extraneous to the linguistic information to be decoded by speech recognizers is caused by the acoustic environment. This includes

background noise, room reverberation, the channel through which the speech is acquired (such as cellular, land-line, and VoIP), overlapping speech, and Lombard or hyper-articulated speech. The acoustic environment in which the speech is captured and the communication channel through which the speech signal is transmitted prior to its processing represent significant causes of harmful variability that is responsible for the drastic degradation of system performance. Existing techniques are able to reduce variability caused by additive noise or linear distortions, as well as compensate for slowly varying linear channels (Droppo and Acero, 2008; Yu et al., 2008). However, more complex channel distortions such as reverberation or fast changing noise, as well as the Lombard effect present a significant challenge to be overcome in future research.

Another common type of speech variability that has been studied intensively is due to different speakers' characteristics. It is well known that speech characteristics vary widely among speakers due to many factors, including speaker physiology, speaker style, and accents—both regional and nonnative. The primary method currently used for making speech-recognition systems more robust to variations in speaker characteristics is to include a wide range of speakers (and speaking styles) in the training. Speaker adaptation mildly alleviates problems with new speakers within the “span” of known speaker/speaking types, but fail for new types. Further, current speech-recognition systems assume a pronunciation lexicon that models native speakers of a language and train on large amounts of speech data from various native speakers of the language. As discussed in the preceding section, a number of modeling approaches have been explored in modeling accented speech, including the explicit modeling of accented speech, the adaptation of native acoustic models via accented speech data with only moderate. Pronunciation variants have also been incorporated in the lexicon to accommodate accented speech, but except for small gains, the problem is largely unsolved. Similarly, some progress has been made for automatically detecting speaking rate from the speech signal, but such knowledge has not been exploited in speech-recognition systems, mainly due to the lack of any explicit mechanism to model speaking rate in the recognition process. Further research is needed to accommodate speaker-related variability.

AQ9

The third common type of speech variability is related to language characteristics including sublanguage or dialect, vocabulary, and genre or topic of conversation. Many important aspects of speaker variability have to do with nonstandard dialects. Dialectal differences in a language can occur in all linguistic aspects: lexicon, grammar (syntax and morphology), and phonology. The vocabulary and language-use in a speech-recognition task change significantly from task to task, necessitating the estimation of new language models for each case. A primary reason language models in current speech-recognition systems are not portable across tasks even within the same language or dialect is that they lack linguistic sophistication—they cannot consistently distinguish meaningful sentences from meaningless ones, nor grammatical from ungrammatical ones. Discourse structure is not considered either, merely the local collocation of words. Another reason why language model adaptation to new domains and genre is very data intensive is the “nonparametric” nature of the current models.

The technical challenge in this area of research will entail the creation and development of systems that would be much more robust against all kinds of variability discussed above, including changes in acoustic environments, reverberation, external noise sources, and communication channels. New techniques and architectures need to be developed to enable exploring these critical issues in meaningful environments as diverse as meeting room presentations to unstructured conversations.

It is fair to say that the acoustic models used in today's speech systems have few explicit mechanisms to accommodate most of the underlying causes of variability described above. The statistical components of the model, such as Gaussian mixtures and Markov chains in the HMM, are instead burdened with implicitly modeling the variability using different mixture components and HMM states and in a frame-by-frame manner. Consequently, when the speech presented to a system deviates along one of these axes from the speech used for parameter estimation, predictions by the models become highly suspect. The performance of the technology degrades catastrophically even when the deviations are such that the intended human listener exhibits little or no difficulty in extracting the same information. The robustness of speech recognition against all these variability factors constitutes a major technical challenge in the field.

The hope for meeting this challenge lies not only in innovative architectures and techniques/algorithms that can intelligently represent explicit mechanisms for the real nature of speech variability, but perhaps more importantly, also in the ever-increasing data available to train and adapt the speech-recognition models in ways not feasible in the past.

### 15.5.2 Capitalizing on Data Deluge for Speech Recognition

We now have some very exciting opportunities to collect large amounts of audio data that have not previously been available. This gives rise to “data deluge.” Thanks in large part to the Internet, there are now readily accessible large quantities of “everyday” speech, reflecting a variety of materials and environments previously unavailable. Other rich sources are university course lectures, seminars, and similar material, which are progressively being put online. All these materials reflect a less formal, more spontaneous, and natural form of speech than present-day systems have typically been developed to recognize. Recently emerging voice search in mobile phones has also provided a rich source of speech data, which, because of the recording of the mobile phone users’ selection, can be considered as partially “labeled.”

One practical benefit of working with these new speech materials is that systems will become more capable and more robust in expanding the range of speech materials that can be accurately recognized under a wide range of conditions. Much of what is learned here is also likely to be of benefit in recognizing casual “everyday” speech in non-English languages.

Over the years, the availability of both open source and commercial speech tools has been very effective in quickly bringing good quality speech processing capabilities to many labs and researchers. New Web-based tools could be made available to collect, annotate, and then process substantial quantities of speech very cost-effectively in many languages. Mustering the assistance of interested individuals on the World Wide Web (e.g., open source software, Wikipedia, etc.) could generate substantial quantities of language resources very efficiently and cost-effectively. This could be especially valuable for creating significant new capabilities for resource “impoverished” languages.

The ever-increasing amount of data, which is increasingly available to help build speech-recognition systems, presents both an opportunity and a challenge for advancing the state of the art in speech recognition. Large corpora of diverse speech will have to be compiled containing speech that carries information of the kind targeted for extraction by speech recognition. It should also exhibit large but usefully normalized extraneous deviations of the kind against which robustness is sought, such as a diverse speaker population with varying degrees of nonnative accents or different local dialects, widely varying channels and acoustic environments, diverse genre, etc.

Speech-recognition technology has barely scratched the surface in sampling the many kinds of speech, environments, and channels that people routinely experience. In fact, we currently provide to our automatic systems only a very small fraction of the amount of materials that humans utilize to acquire language. If we want our systems to be more powerful and to understand the nature of speech itself, we need to make more use of it and label more of it. Well-labeled speech corpora have been the cornerstone on which today’s systems have been developed and evolved. However, most of the large quantities of data are not labeled or poorly “labeled,” and labeling them accurately is costly. There is an urgent and practical need to develop high-quality active learning and unsupervised/semi-supervised learning techniques. Upon their successful development, the exploitation of unlabeled or partially labeled data becomes possible to train the models, and we can automatically (and “actively”) select parts of the unlabeled data for manual labeling in a way that maximizes its utility. This need is partly related to the compilation of diverse training data discussed earlier. The range of possible combinations of channel, speaker, environment, speaking style, and domain is so large that it is unrealistic to expect transcribed or labeled speech in every configuration of conditions for training the models. However, it is feasible to simply collect raw speech in all conditions of interest. Another important reason for unsupervised learning is that the systems,

like their human “baseline,” will have to undergo “lifelong learning,” adjusting to evolving vocabulary, channels, language use, etc.

Large amounts of speech data will enable multi-stream and multiple-module strategies for speech recognition to be developed. Robust methods are needed to identify reliable elements of the speech spectrum in a data-driven manner by employing an entire ensemble of analyses. A multiple-module approach also entails a new search strategy that treats the reliability of a module or stream in any instance as another hidden variable over which to optimize, and seeks the most likely hypothesis over all configurations of these hidden variables.

### 15.5.3 Self-Learning and Adaptation for Speech Recognition

State-of-the-art systems for speech recognition are based on statistical models estimated from labeled training data, such as transcribed speech, and from human-supplied knowledge, such as pronunciation dictionaries. Such built-in knowledge often becomes obsolete fairly quickly after a system is deployed in a real-world application, and significant and recurring human intervention in the form of retraining is needed to sustain the utility of the system. This is in sharp contrast with the speech facility in humans, which is constantly updated over a lifetime, routinely acquiring new vocabulary items and idiomatic expressions, as well as deftly handling previously unseen nonnative accents and regional dialects of a language. In particular, humans exhibit a remarkable aptitude for learning the sublanguage of a new domain or application without explicit supervision.

The challenge here is to create self-adaptive or self-learning techniques that will endow speech recognizers with at least a rudimentary form of the human’s self-learning capability. There is a need for learning at all levels of speech and language processing to cope with changing environments, nonspeech sounds, speakers, pronunciations, dialects, accents, words, meanings, and topics, to name but a few sources of variation over the lifetime of a deployed system. Like its human counterpart, the system would engage in automatic pattern discovery, active learning, and adaptation. Research in this area must address both the learning of new models and the integration of such models into preexisting knowledge sources. Thus, an important aspect of learning is being able to discern when something has been learned and how to apply the result. Learning from multiple concurrent modalities, for example, new text and video, may also be necessary. For instance, a speech-recognition system may encounter a new proper noun in its input speech, and may need to examine contemporaneous text with matching context to determine the spelling of the name. The exploitation of unlabeled or partially labeled data would be necessary for such learning, perhaps including the automatic selection (by the system) of parts of the unlabeled data for manual labeling, in a way that maximizes its utility.

A motivation for the research direction on developing speech recognizers’ self-learning capability is the growing activity in the allied field of Machine Learning. Success in this endeavor would extend the lifetime of deployed systems, and directly advance our ability to develop speech systems in new languages and domains without onerous demands of labeled speech, essentially by creating systems that automatically learn and improve over time.

One most important aspect of learning is generalization. When a small amount of test data is available to adjust speech recognizers, we call such generalization as adaptation. Adaptation and generalization capabilities enable rapid speech-recognition application integration.

Over the past three decades, the speech community has developed and refined an experimental methodology that has helped to foster steady improvements in speech technology. The approach that has worked well, and been adopted in other research communities, is to develop shared corpora, software tools, and guidelines that can be used to reduce differences between experimental setups down to the basic algorithms, so that it becomes easier to quantify fundamental improvements. Typically, these corpora are focused on a particular task. As speech technology has become more sophisticated, the scope and difficulty of these tasks has continually increased: from isolated words to continuous speech, from speaker-dependent to independent, from read to spontaneous speech, from clean to noisy, from utterance

to content-based, etc. Although the complexity of such corpora has continually increased, one common property of such tasks is that they typically have a training partition that is quite similar in nature to the test data. Indeed, obtaining large quantities of training data that is closely matched to the test is perhaps the single most reliable method to improve speech-recognition performance. This strategy is quite different from the human experience however. For our entire lives, we are exposed to all kinds of speech data from uncontrolled environments, speakers, and topics, (i.e., “every day” speech). Despite this variation in our own personal training data, we are all able to create internal models of speech and language that are remarkably adept at dealing with variation in the speech chain. This ability to generalize is a key aspect of human speech processing that has not yet found its way into modern speech recognizers. Research activities on this topic should produce technology that will operate more effectively in novel circumstances, and that can generalize better from smaller amounts of data. Examples include moving from one acoustic environment to another, different tasks, languages, etc. Another research area could explore how well information gleaned from large resource languages and/or domains generalize to smaller resource languages and domains.

#### 15.5.4 Developing Speech Recognizers beyond the Language Barrier

State-of-the-art speech recognition systems today deliver top performances by building complex acoustic and language models using a large collection of domain- and language-specific speech and text examples. This set of language resources is often not readily available for many languages. The challenge here is to create spoken language technologies that are rapidly portable. To prepare for rapid development of such spoken language systems, a new paradigm is needed to study speech and acoustic units that are more language-universal than language-specific phones. Three specific research issues need to be addressed: (1) cross-language acoustic modeling of speech and acoustic units for a new target language, (2) cross-lingual lexical modeling of word pronunciations for new language, (3) cross-lingual language modeling. By exploring correlation between these emerging languages and well-studied languages, cross-language properties (e.g., language clustering and universal acoustic modeling can be utilized to facilitate the rapid adaptation of acoustic and language models. Bootstrapping techniques are also keys to building preliminary systems from a small amount of labeled utterances first, using them to label more utterance examples in an unsupervised manner, incorporating new-labeled data into the label set, and iterating to improve the systems until they reach a comparable performance level similar to today’s high-accuracy systems.

#### 15.5.5 Detection of Unknown Events in Speech Recognition

Current ASR systems have difficulty in handling unexpected—and thus often the most information rich—lexical items. This is especially problematic in speech that contains interjections or foreign or out-of-vocabulary words, and in languages for which there is relatively little data with which to build the system’s vocabulary and pronunciation lexicon. A common outcome in this situation is that high-value terms are overconfidently misrecognized as some other common and similar-sounding word. Yet, such spoken events are key to tasks such as spoken term detection and information extraction from speech. Their accurate detection is therefore of vital importance.

The challenge here is to create systems that reliably detect when they do not know a (correct) word. A clue to the occurrence of such error events is the mismatch between an analysis of a purely sensory signal unencumbered by prior knowledge, such as unconstrained phone recognition, and a word- or phrase-level hypothesis based on higher level knowledge, often encoded in a language model. A key component of this research would therefore be to develop novel confidence measures and accurate models of uncertainty based on the discrepancy between sensory evidence and *a priori* beliefs. A natural sequel to detection of such events would be to transcribe them phonetically when the system is confident that its word hypothesis is unreliable, and to devise error-correction schemes.

### 15.5.6 Learning from Human Speech Perception and Production

As a long-term research direction, one principal knowledge source that we can draw to benefit machine speech recognition is in the area of human speech perception, understanding, and cognition. This rich knowledge source has its basis in both psychological and physiological processes in humans. Physiological aspects of the human speech perception of most interest include cortical processing in the auditory area as well as in the associated motor area of the brain. One important principle of auditory perception is its modular organization, and recent advances in functional neuroimaging technologies provide a driving force motivating new studies toward developing the integrated knowledge of the modularly organized auditory process in an end-to-end manner. The psychological aspects of human speech perception embody the essential psychoacoustic properties that underlie auditory masking and attention. Such key properties equip human listeners with the remarkable capability of coping with cocktail party effects that no current automatic speech-recognition techniques can successfully handle. Intensive studies are needed in order for speech recognition and understanding applications to reach a new level, delivering performance comparable to humans.

Specific issues to be resolved in the study of how the human brain processes spoken (as well as written) language are the way human listeners adapt to nonnative accents and the time course over which human listeners reacquaint themselves to a language known to them. Humans have amazing capabilities to adapt to nonnative accents. Current speech-recognition systems are extremely poor in this aspect, and the improvement is expected only after we have sufficient understanding of human speech processing mechanisms. One specific issue related to human speech perception, which is linked to human speech production, is the temporal span over which speech signals are represented and modeled. One prominent weakness in current HMMs is the handicap in representing long-span temporal dependency in the acoustic feature sequence of speech, which, nevertheless, is an essential property of speech dynamics in both perception and production. The main cause of this handicap is the conditional independence assumptions inherit in the HMM formalism. The HMM framework also assumes that speech can be described as a sequence of discrete units, usually phone(me)s. In this symbolic, invariant approach, the focus is on the linguistic/phonetic information, and the incoming speech signal is normalized during preprocessing in order to remove most of the paralinguistic information. However, human speech perception experiments have shown that the paralinguistic information plays a crucial role in human speech perception.

Numerous approaches have been taken over the past dozen years to address the above weaknesses of HMMs. These approaches can be broadly classified into the following two categories. The first, parametric, structure-based approach establishes mathematical models for stochastic trajectories/segments of speech utterances using various forms of parametric characterization. The essence of such an approach is that it exploits knowledge and mechanisms of human speech perception and production so as to provide the structure of the multitiered stochastic process models. These parametric models account for the observed speech trajectory data based on the underlying mechanisms of speech coarticulation and reduction directly relevant to human speech perception, and on the relationship between speaking rate variations and the corresponding changes in the acoustic features. The second, nonparametric and template-based approach to overcoming the HMM weaknesses involves the direct exploitation of speech feature trajectories (i.e., “template”) in the training data without any modeling assumptions. This newer approach is based on episodic learning as evidenced in many recent human speech perception and recognition experiments. Due to the dramatic increase of speech databases and computer storage capacity available for training, as well as the exponentially expanded computational power, nonparametric methods and episodic learning provide rich areas for future research. The essence of the template-based approach is that it captures strong dynamic segmental information about speech feature sequences in a way complimentary to the parametric, structure-based approach.

Understanding human speech perception will provide a wealth of information enabling the construction of better models (than HMMs) that reflect attributes of human auditory processing and the linguistic units

used in human speech recognition. For example, to what extent may human listeners use mixed word or phrase “templates” and the constituent phonetic/phonological units in their memory to achieve relatively high-performance in speech recognition for accented speech or foreign languages (weak knowledge) and for acoustically distorted speech (weak observation)? How do human listeners use episodic learning (e.g., direct memory access) and parametric learning related to smaller phonetic units (analogous to what we are currently using for HMMs in machines) in speech recognition/understanding? Answers to these questions will benefit our design of next-generation machine speech-recognition models and algorithms.

### **15.5.7 Capitalizing on New Trends in Computational Architectures for Speech Recognition**

Moore’s law has been a dependable indicator of the increased capability for computation and storage in our computational systems for decades. The resulting effects on systems for speech recognition and understanding have been enormous, permitting the use of larger and larger training databases and recognition systems, and the incorporation of more and more detailed models of spoken language. Many of the future research directions and applications suggested in this chapter implicitly depend upon a continued advance in computational capabilities, an assumption that certainly seems justified given recent history. However, the fundamentals of this progression have recently changed. As Intel and others have noted recently, the power density on microprocessors has increased to the point that higher clock rates would begin to melt the silicon. Consequently, at this point, industry development is now focused on implementing microprocessors on multiple cores. Dual core CPUs are now very common, and four-processor and eight-processor systems are coming out. The new road maps for the semiconductor industry reflect this trend, and future speedups will come more from parallelism than from having faster individual computing elements.

For the most part, algorithm designers for speech systems have ignored investigation of such parallelism, partly because the advance of scalability has been so reliable. Future research directions and applications discussed in this chapter will require significantly more computation, and consequently researchers concerned with implementation will need to consider parallelism explicitly in their designs. This will be a significant change from the status quo. In particular, tasks such as decoding, for which extremely clever schemes to speed up single-processor performance have been developed, will require a complete rethinking of the algorithms.

### **15.5.8 Embedding Knowledge and Parallelism into Speech-Recognition Decoding**

Decoding or search is one of the three major components in the general statistical speech-recognition architecture, as we overviewed earlier in this chapter on the conventional techniques developed including the time-synchronous Viterbi search and the stack search. These search algorithms were developed long before parallelism came into being. New search methods that explicitly exploit parallelism as a novel computational architecture may be an important research direction for speech understanding systems.

Additionally, as innovative recognition algorithms are added, there will be impact on the search component. For instance, rather than the left-to-right (and sometimes right-to-left) recognition passes that are used today, there could be advantages to either identifying islands of reliability or islands of uncertainty, and rely upon alternate knowledge sources only “locally” in the search process. The incorporation of multiple tiers of units (e.g., Deng and Sun, 1994; Sun and Deng, 2002), such as articulatory feature, sub-phone state, phone, syllable, word, and multi-word phrase, could have consequences for the search process.

Further, so-called episodic approaches to speech recognition are being investigated (Wachter et al., 2003). These approaches rely on examples of phrases, words, or other units directly, as opposed to statistical models of speech. While this seems to be a throwback to the days before the prominence of

HMMs, the idea is gaining new prominence due to the availability of larger and larger speech databases, and thus more and more examples for each modeled speech unit (Deng and Strik, 2007). One future research direction would be to learn how to best incorporate these approaches into a search that also uses statistical models, which have already proven their worth.

## 15.6 Summary

---

Speech recognition has a long history of development. It is not until the introduction of the statistical framework that the field has enjoyed steadfast progress and has opened up many practical applications. Three main components (acoustic modeling, language modeling, and decoding) discussed in this chapter can be found in most modern speech-recognition systems. Each of these components has had significant milestones in the historical developments. Beyond the set of applications discussed in this chapter, we believe other speech applications will proliferate thanks to the increased power of computing, mobile communications, and multimodal user interface, as well as to new breakthroughs and research advances in the field. Some of the fertile areas for future research discussed in Section 15.5 provide potential advances that may lead to new speech technology applications.

## References

---

AQ10

- Anastasakos, T., J. McDonough, and J. Makhoul (1997). Speaker adaptive training: A maximum likelihood approach to speaker normalization, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1043–1046, Munich, Germany.
- Bahl, L., P. Brown, P. de Souza, and R. Mercer (1986). Maximum mutual information estimation of hidden Markov model parameters for speech recognition, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 49–52, Tokyo, Japan.
- Baker, J. (1975). Stochastic modeling for automatic speech recognition, in D. R. Reddy, (ed.), *Speech Recognition*, Academic Press, New York.
- Baker, J., L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, and N. Morgan (2007). MINDS report: Historical development and future directions in speech recognition and understanding. <http://www-nlp.nist.gov/MINDS/FINAL/speech.web.pdf>.
- Baker, J., L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. O'Shaughnessy (2009a). Updated MINDS report on speech recognition and understanding part I, *IEEE Signal Processing Magazine*, 26(3), 75–80.
- Baker, J., L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. O'Shaughnessy (2009b). Updated MINDS report on speech recognition and understanding part II, *IEEE Signal Processing Magazine*, 26(4).
- Baum, L. (1972). An inequality and associated maximization technique occurring in statistical estimation for probabilistic functions of a Markov process, *Inequalities*, III, 1–8.
- Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984). *Classification and Regression Trees*, Wadsworth & Brooks, Pacific Grove, CA.
- Chelba C. and F. Jelinek (2000). Structured language modeling, *Computer Speech and Language*, 14, 283–332.
- Davis, S. and P. Mermelstein (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), 357–366.
- Dempster, A., N. Laird, and D. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*, 39(1), 1–21.

AQ11

AQ12

- Deng, L. (1993). A stochastic model of speech incorporating hierarchical nonstationarity, *IEEE Transactions on Speech and Audio Processing*, 1(4), 471–475.
- Deng, L., M. Aksmanovic, D. Sun, and J. Wu (1994). Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states, *IEEE Transactions on Speech and Audio Processing*, 2, 507–520.
- Deng, L. and X. D. Huang (2004). Challenges in adopting speech recognition, *Communications of the ACM*, 47(1), 11–13.
- Deng, L. and D. O'Shaughnessy (2003). *Speech Processing—A Dynamic and Optimization-Oriented Approach*, Marcel Dekker Inc., New York.
- Deng, L. and H. Strik (2007). Structure-based and template-based automatic speech recognition—Comparing parametric and non-parametric approaches, in *Proceedings of the 8th Annual Conference of the International Speech Communication Association Interspeech*, Antwerp, Belgium.
- Deng, L. and D. Sun (1994). A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features, *Journal of the Acoustical Society of America*, 85(5), 2702–2719.
- Deng, L., D. Yu, and A. Acero (2006). Structured speech modeling, *IEEE Transactions on Audio, Speech and Language Processing (Special Issue on Rich Transcription)*, 14(5), 1492–1504.
- Droppo, J. and A. Acero (2008). Environmental robustness, in *Handbook of Speech Processing*, Springer-Verlag, Berlin, Germany.
- Eide E. and H. Gish (1996). A parametric approach to vocal tract length normalization, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 346–349, Atlanta, GA.
- Fiscus, J. (1997). A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER), *IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 3477–3482, Santa Barbara, CA.
- Furui, S. (2001). *Digital Speech Processing, Synthesis and Recognition* (2nd Ed.), Marcel Dekker Inc., New York.
- Gao Y. and J. Kuo (2006). Maximum entropy direct models for speech recognition, *IEEE Transactions on Speech and Audio Processing*, 14(3), 873–881.
- Gauvain, J.-L. and C.-H. Lee (1997). Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, *IEEE Transactions on Speech and Audio Processing*, 7, 711–720.
- Glass, J. (2003). A probabilistic framework for segment-based speech recognition, in M. Russell and J. Bilmes (eds.), *New Computational Paradigms for Acoustic Modeling in Speech Recognition, Computer, Speech and Language* (Special issue), 17(2–3), 137–152.
- Gold, B. and N. Morgan (2000). *Speech and Audio Signal Processing*, John Wiley & Sons, New York.
- Gunawardana, A. and W. Byrne (2001). Discriminative speaker adaptation with conditional maximum likelihood linear regression, *Proceedings of the EUROSPEECH*, Aalborg, Denmark.
- He, X., L. Deng, C. Wu (2008). Discriminative learning in sequential pattern recognition, *IEEE Signal Processing Magazine*, 25(5), 14–36.
- Hermansky, H. (1990). Perceptual linear predictive analysis of speech, *Journal of the Acoustical Society of America*, 87(4), 1738–1752.
- Hermansky H. and N. Morgan (1994). RASTA processing of speech, *IEEE Transactions on Speech and Audio Processing*, 2(4), 578–589.
- Huang, X. D. (2009). Leading a start-up in an enterprise: Lessons learned in creating Microsoft response point, *IEEE Signal Processing Magazine*, 26(2).
- Huang, X. D., A. Acero, and H. Hon (2001). *Spoken Language Processing—A Guide to Theory, Algorithms, and System Development*, Prentice Hall, Upper Saddle River, NJ.
- Huang, X. D. and K.-F. Lee (1993). On speaker-independent, speaker-dependent and speaker adaptive speech recognition, *IEEE Transactions on Speech and Audio Processing*, 1(2), 150–157.
- Jelinek, F. (1969). A fast sequential decoding algorithm using a stack, *IBM Journal of Research and Development*, 13, 675–685.

AQ13

- Jelinek, F. (1976). Continuous speech recognition by statistical methods, *Proceedings of the IEEE*, 64(4), 532–557.
- Jelinek, F. (1997). *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, MA.
- Jiang, L. and X. D. Huang (1998). Vocabulary-independent word confidence measure using subword features, in *Proceedings of the International Conference on Spoken Language Processing*, pp. 401–404, Sydney, NSW.
- Jurafsky D. and J. Martin (2000). *Speech and Language Processing—An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, Upper Saddle River, NJ.
- Kumar, N. and A. Andreou (1998) Heteroscedastic analysis and reduced rank HMMs for improved speech recognition, *Speech Communication*, 26, 283–297.
- Lee, K. F. (1988). *Automatic Speech Recognition: The Development of the Sphinx Recognition System*, Springer-Verlag, Berlin, Germany.
- Lee, C., F. Soong, and K. Paliwal (eds.) (1996). *Automatic Speech and Speaker Recognition—Advanced Topics*, Kluwer Academic, Norwell, MA.
- Leggetter C. and P. Woodland (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models, *Computer Speech and Language*, 9, 171–185.
- Lippman, R. (1987). An introduction to computing with neural nets, *IEEE ASSP Magazine*, 4(2), 4–22.
- Macherey, M., L. Haferkamp, R. Schlüter, and H. Ney (2005). Investigations on error minimizing training criteria for discriminative training in automatic speech recognition, in *Proceedings of Interspeech*, pp. 2133–2136, Lisbon, Portugal.
- Morgan, N., Q. Zhu, A. Stolcke, K. Sonmez, S. Sivadas, T. Shinozaki, M. Ostendorf, P. Jain, H. Hermansky, D. Ellis, G. Doddington, B. Chen, O. Cetin, H. Bourlard, and M. Athineos (2005). Pushing the envelope—Aside, *IEEE Signal Processing Magazine*, 22, 81–88.
- Ney, H. (1984). The use of a one-stage dynamic programming algorithm for connected word recognition, *IEEE Transactions on ASSP*, 32, 263–271.
- Ostendorf, M., V. Digalakis, and J. Rohlicek (1996). From HMMs to segment models: A unified view of stochastic modeling for speech recognition, *IEEE Transactions on Speech and Audio Processing*, 4, 360–378.
- Poritz, A. (1998). Hidden Markov models: A guided tour, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 1–4, Seattle, WA.
- Povey, B., Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig (2005). FMPE: Discriminatively trained features for speech recognition, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA.
- Rabiner, L. and B. Juang (1993). *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, NJ.
- Reddy, D. R. (ed.) (1975). *Speech Recognition*, Academic Press, New York.
- Rosenberg, A., C. H. Lee, and F. K. Soong (1994). Cepstral channel normalization techniques for HMM-based speaker verification, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1835–1838, Adelaide, SA.
- Sakoe, S. and S. Chiba (1971). A dynamic programming approach to continuous speech recognition, in *Proceedings of the 7th International Congress on Acoustics*, Vol. 3, pp. 65–69, Budapest, Hungary.
- Vintsyuk, T. (1968). Speech discrimination by dynamic programming, *Kibernetika*, 4(2), 81–88.
- Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, in *IEEE Transactions on Information Theory*, IT-13(2), 260–269.
- Schwartz, R. and Y. Chow (1990). The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, NM.

AQ14

- Sun, J. and L. Deng (2002). An overlapping-feature based phonological model incorporating linguistic constraints: Applications to speech recognition, *Journal of the Acoustical Society of America*, 111(2), 1086–1101.
- Vinyals, O., L. Deng, D. Yu, and A. Acero (2009) Discriminative pronunciation learning using phonetic decoder and minimum-classification-error criterion, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan.
- Wang, Y., M. Mahajan, and X. Huang (2000). A unified context-free grammar and n-gram model for spoken language processing, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey.
- Wang, Y., D. Yu, Y. Ju, and A. Acero (2008). An introduction to voice search, *IEEE Signal Processing Magazine (Special Issue on Spoken Language Technology)*, 25(3), 29–38.
- Wachter, M., K. Demuynck, D. Van Compernolle, and P. Wambacq (2003). Data-driven example based continuous speech recognition, in *Proceedings of the EUROSPEECH*, pp. 1133–1136, Geneva.
- Yaman, S., L. Deng, D. Yu, Y. Wang, and A. Acero (2008). An integrative and discriminative technique for spoken utterance classification, *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6), 1207–1214.
- Yu, D., L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero (2008). Robust speech recognition using cepstral minimum-mean-square-error noise suppressor, *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5).

AQ15

## AUTHOR QUERIES

- [AQ1] O'Shaughnessy (2000) is not provided in the reference list. please check.
- [AQ2] Poritz (1988) is not provided in the reference list. please check.
- [AQ3] Gunawardana et al. (2006) is not provided in the reference list. please check.
- [AQ4] Juang et al. (1997) is not provided in the reference list. please check.
- [AQ5] He et al. (2008) is not provided in the reference list. please check.
- [AQ6] Deng et al. (1993) is not provided in the reference list. please check.
- [AQ7] Gauvain and Lee (1994) is not provided in the reference list. please check.
- [AQ8] Figures 15.6 through 15.9 are renumbered for sequential appearance in the text. Please check.
- [AQ9] The sentence “As discussed...only moderate.” is incomplete. Please check.
- [AQ10] Please provide in text citation for Gauvain and Lee (1997); Gunawardana and Byrne (2001); Huang (2009); Jiang and Huang (1998); Poritz (1998); Reddy (1975); Yaman et al. (2008); Vinyals et al. (2009); and Breiman et al. (1984).
- [AQ11] Please provide page range for Baker et al. (2009b).
- [AQ12] Please provide editor names and page range for Droppo and Acero (2008).
- [AQ13] Please provide page range for Huang (2009).
- [AQ14] Please check the publisher name and location in Lee (1988).
- [AQ15] Please provide page range for Yu et al. (2008).