12 Speech Recognition

STEVE RENALS AND THOMAS HAIN

1 Introduction

By *automatic speech recognition* we mean the process of speech-to-text transcription: the transformation of an acoustic signal into a sequence of words, without necessarily understanding the meaning or intent of what was spoken. Recognition without understanding is not always possible since some semantic context may be required, for instance, to disambiguate *Will the new display recognize speech?* from *Will the nudist play wreck a nice beach?* Automatic speech recognition corresponds to answering the question *who spoke what when?*;[1] in the general case this may involve transcribing the speech of two or more people taking part in a conversation, in which the speakers frequently talk at the same time. The solution to this task is often considered in two parts: *speaker diarization* (who spoke when?), and *speech-to-text transcription* (what did they say?). Of course, the speech signal contains much more information than just the words spoken and who said them. Speech acoustics also carries information about timing, intonation, and voice quality. These paralinguistic aspects convey information about the speaker's emotion and physiology, as well as sometimes disambiguating between different possible meanings. In this chapter, however, we shall focus on automatic speech-to-text transcription.

Speech-to-text transcription has a number of applications including the dictation of office documents, spoken dialogue systems for call centers, hands-free interfaces to computers, and the development of speech-to-speech translation systems. Each of these applications is typically more restricted than the general problem which requires the automatic transcription of naturally spoken continuous speech, by an unknown speaker in any environment. This is an extremely challenging task. There are several sources of variability which we cluster into four main areas: the *task domain*, *speaker characteristics*, *speaking style*, and the recognition *environment*. In many practical situations, the variability is restricted. For example, there may be a single, known speaker, or the speech to be recognized may be carefully dictated text rather than a spontaneous conversation, or the recording environment may be quiet and non-reverberant. In speech-to-text transcription a distinction is made between parts addressing acoustic variability (acoustic modeling), and parts addressing linguistic uncertainty (language modeling).

1.1.1 Task domain Aspects of the specific speech recognition task which affect the difficulty of the speech transcription process include the language and the size of the vocabulary to be recognized, and whether the speech comes from a limited domain. Different languages present different challenges for a speech recognizer. For example, agglutinative languages, such as Turkish and Finnish, have larger vocabularies than non-agglutinative languages (such as English) due to words formed from the concatenation of multiple morphemes. This has an effect on the lexical and language model. For English, a speech recognition system is considered to have a large vocabulary if it has the order of 10^4 word types in its vocabulary; a comparable system for a language such as Finnish, however, may have two orders of magnitude more word types. As another example, tonal languages, such as Chinese, use pitch movements to distinguish small sets of words, which has an effect on acoustic modeling.

The number of word types in the vocabulary of a speech recognizer gives an indication of the 'size' of the problem that may be misleading, however; perplexity[2] of the language model gives an indication of effective size. A spoken dialogue system concerned with stock prices, for instance, may have a relatively large vocabulary (due to the number of distinct words occurring in company names) but a small perplexity.

1.1.2 Speaker characteristics Different speakers have differences in their speech production anatomy and physiology, speak at different rates, use different language, and produce speech acoustics with different degrees of intrinsic variability. Other differences in speaker characteristics arise from systematic variations such as those arising from speaker age and accent. One way to deal with this variability is through the construction of *speaker-dependent* speech recognition systems, but this demands a new system to be constructed for each speaker. *Speaker-independent* systems, on the other hand, are more flexible in that they are designed to recognize any speaker. In practice, a speaker-dependent speech recognition system will tend to make fewer errors than a speaker-independent system. Although speaker adaptation algorithms (Section 2.5) have made great progress over the past 15 years, it is still the case that the adaptability and robustness to different speakers exhibited by automatic speech recognition systems is very limited compared with human performance.

1.1.3 Style In the early days of automatic speech recognition, systems solved the problem of where to locate word boundaries by requiring the speaker to leave pauses between words: the pioneering dictation product Dragon Dictate (Baker 1989) is a good example of a large-vocabulary *isolated word* system. However, this is an unnatural speaking style and most research in speech recognition is focused on *continuous* speech recognition, in which word boundary information is not easily available. The problem of continuous speech recognition thus involves segmentation into words, as well as labeling each word.

Until the mid-1990s most speech recognition research followed research in acoustic phonetics, using recordings of planned speech recorded in laboratory or quiet office conditions. However, it has become apparent that more natural styles of speech, as observed in spontaneous conversation, result in considerably more acoustic variability: this is reflected in the increased word error rates for *conversational* or *spontaneous* speech recognition compared with the recognition of dictation or other *planned* speech. A modern speech recognition system for the transcription of dictated newspaper text results in a typical word error rate of 5–10 percent; a state-of-the-art conversational speech recognition system will result in a word error rate of 10–30 percent when transcribing spontaneous conversations (Chen et al., 2006b; Hain et al., 2007).

1.1.4 Environment Finally the acoustic environment in which the speech is recorded, along with any transmission channel can have a significant impact on the accuracy of a speech recognizer. Outside of quiet offices and laboratories, there are usually multiple acoustic sources including other talkers, environmental noise and electrical or mechanical devices. In many cases, it is a significant problem to separate the different acoustic signals found in an environment. In addition, the microphone on which the speech is recorded may be close to the talker (in the case of a headset or telephone), attached to a lapel, or situated on a wall or tabletop. Variations in transmission channel occur due to movements of the talker's head relative to the microphone and transmission across a telephone network or the internet. Probably the largest disparity between the accuracy of automatic speech recognition compared with human speech recognition occurs in situations with high additive noise, multiple acoustic sources, or reverberant environments.

1.2 *Learning from data*

The standard framework for speech recognition is statistical, developed in the 1970s and 1980s by Baker (1975), a team at IBM (Jelinek 1976; Bahl et al., 1983), and a team at AT&T (Levinson et al., 1983; Rabiner 1989). In this formulation the most probable sequence of words $\mathbf{W}^*$ must be identified given the recorded acoustics $\mathbf{X}$ and the model $\boldsymbol{\theta}$:

(1) $$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X}, \boldsymbol{\theta})$$

(2) $$= \arg\max_{\mathbf{W}} \frac{p(\mathbf{X} \mid \mathbf{W}, \boldsymbol{\theta}) P(\mathbf{W} \mid \boldsymbol{\theta})}{p(\mathbf{X} \mid \boldsymbol{\theta})}$$

(3) $$= \arg\max_{\mathbf{W}} p(\mathbf{X} \mid \mathbf{W}, \boldsymbol{\theta}) P(\mathbf{W} \mid \boldsymbol{\theta})$$

(4) $$= \arg\max_{\mathbf{W}} \log p(\mathbf{X} \mid \mathbf{W}, \boldsymbol{\theta}) + \log P(\mathbf{W} \mid \boldsymbol{\theta})$$

Equation (1) specifies the most probable word sequence as the one with the highest posterior probability given the acoustics and the model. Equation (2) follows from (1) through the application of Bayes's theorem; since $p(\mathbf{X} \mid \boldsymbol{\theta})$ is independent of the word sequence, we usually work with (3), or its log-domain version (4). P denotes a probability and p denotes a probability density function (pdf). In what follows, the dependence on the model $\boldsymbol{\theta}$ (which is usually fixed) is suppressed to avoid notational clutter.

Equations (3) or (4) may be regarded as splitting the problem into two components: *language modeling*, which is concerned with estimating the prior probability of a word sequence $P(\mathbf{W})$, and *acoustic modeling*, in which the likelihood of the acoustic data given the words, $p(\mathbf{X} \mid \mathbf{W})$, is estimated. The parameters of both of these models are normally learned from large annotated corpora of data. Obtaining the optimal word sequence $\mathbf{W}^*$ is the *search* or *decoding* problem, discussed further in section 3.

The language model $P(\mathbf{W})$, which is discussed further in Chapter 3, STATISTICAL LANGUAGE MODELING, models a word sequence by providing a predictive probability distribution for the next word based on a history of previously observed words. Since this probability distribution does not depend on the acoustics, language models may be estimated from large textual corpora. Of course, the statistics of spoken word sequences are often rather different from the statistics of written text. The conventional n-gram language model, which approximates the history as the immediately preceding $n-1$ words, has represented the state of the art for large-vocabulary speech recognition for 25 years. It has proven difficult to improve over this simple model (Jelinek 1991; Rosenfeld, 2000). Attempts to do so have focused on improved models of word sequences (e.g., Bengio et al., 2003; Blitzer et al., 2005; Teh 2006; Huang & Renals 2007) or the incorporation of richer knowledge (e.g., Bilmes & Kirchhoff 2003; Emami & Jelinek 2005; Wallach 2006).

In this chapter we focus on acoustic modeling, and the development of systems for the recognition of conversational speech. In particular we focus on the trainable hidden Markov model/Gaussian mixture model (HMM/GMM) for acoustic modeling, the choice of modeling unit, and issues including adaptation, robustness, and discrimination. We also discuss the construction of a fielded system for the automatic transcription of multiparty meetings.

1.3 *Corpora and evaluation*

The statistical framework for ASR is extremely powerful: it is scalable and efficient algorithms to estimate the model parameters from a corpus of speech data (transcribed at the word level) are available.

The availability of standard corpora, together with agreed evaluation protocols, has been very important in the development of the field. The specification, collection, and release of the TIMIT corpus (Fisher et al., 1986) marked a significant point in the history of speech recognition research. This corpus, which has been widely used by speech recognition researchers for over two decades, contains utterances from 630 North American speakers, and is phonetically transcribed and time-aligned. The corpus defined training and test sets, together with a commonly agreed evaluation metric (phone error rate – analogous to word error rate discussed below). This resulted in a training and evaluation protocol enabling the exact comparison of results between researchers.

Since the release of TIMIT, many speech corpora with corresponding evaluation protocols have been released. These include corpora of domain-specific read speech (e.g., DARPA Resource Management), read aloud newspaper text (e.g., *Wall Street Journal*), domain-specific human–computer dialogues (e.g., ATIS), broadcast news recordings (e.g., Hub4), conversational telephone speech (e.g., Switchboard), and recordings of multiparty meetings (e.g., AMI). Many of these corpora are available from the Linguistic Data Consortium (www.ldc.upenn.edu); the AMI corpus is available from http://corpus.amiproject.org. The careful recording, transcription, and release of speech corpora has been closely connected to a series of benchmark evaluations of automatic speech recognition systems, primarily led by the US National Institute of Standards and Technology (NIST). This cycle of data collection and system evaluation has given speech recognition research a solid objective grounding, and has resulted in consistent improvements in the accuracy of speech recognition systems (Deng & Huang 2004) – although Bourlard et al. (1996), among others, have argued that an overly strong focus on evaluation can lead to a reduction in innovation.

If the speech recognition problem is posed as the transformation of an acoustic signal to a single stream of words, then there is widespread agreement on word error rate (WER) as the appropriate evaluation measure. The sequence of words output by the speech recognizer is aligned to the reference transcription using dynamic programming. The accuracy of the speech recognizer may then be estimated as the string edit distance between the output and reference strings. If there are N words in the reference transcript, and alignment with the speech recognition output results in S substitutions, D deletions, and I insertions, the word error rate is defined as:

(5) $$\text{WER} = 100 \cdot \frac{(S + D + I)}{N} \%$$

(6) $$\text{Accuracy} = (100 - \text{WER})\%$$

In the case of a high number of insertions, it is possible for the WER to be above 100 percent. Computation of WER is dependent on the automatic alignment between the reference and hypothesized sequence of words. As word timings are not used in the process this may lead to underestimates of true error rate in situations of considerable mismatch. In practice, the transition costs used in the dynamic programming algorithm to compute the alignment are standardized and embedded in standard software implementations such as the NIST sclite tool,[3] or the HResults tool in the HTK speech recognition toolkit.[4]

More generally, the desired output of a speech recognition system cannot always be expressed as a single sequence of words. Multiparty meetings, for instance, are characterised by multiple overlapping speakers. A measure of transcription quality for meetings might usefully include attributing each word to a meeting participant, as well as including of timing information, to take account of overlaps.

2 Acoustic Modeling

The statistical formulation of the speech recognition problem outlined in Section 1.2 provides the basic framework for all state-of-the-art systems. The acoustic model, which is used to estimate $p(\mathbf{X} \mid \mathbf{W})$, may be interpreted as a generative model of a word sequence. Such a model must be decomposable into smaller units, since it is infeasible to estimate a separate model for each word sequence. Hidden Markov models (HMMs) (Baker 1975; Poritz 1988; Rabiner 1989; Jelinek 1998) have proven to be very well suited to this task.

HMMs are probabilistic finite state machines, which may be combined hierarchically to construct word sequence models out of smaller units. In large-vocabulary speech recognition systems, word sequence models are constructed from word models, which in turn are constructed from subword models (typically context-dependent phone models) using a pronunciation dictionary.

HMM acoustic models treat the speech signal as arising from a sequence of discrete phonemes, or 'beads-on-a-string' (Ostendorf 1999). Such a modeling approach does not (directly) take into account processes such as *coarticulation*, a phenomenon in which the place of articulation for one speech sound depends on a neighboring speech sound. For instance, consider the phoneme /n/ in the words '*ten*' and '*tenth*.' In '*ten*,' /n/ is dental, with the tongue coming into contact (or close to) the upper front teeth; in '*tenth*,' /n/ is alveolar, with the tongue farther back in the mouth (coming into contact with the alveolar ridge). Coarticulation gives rise to significant context-dependent variability. The use of context-dependent phone modeling (Section 2.3) aims to mitigate these effects, as does the development of richer acoustic models that take account of speech production knowledge (King et al., 2007).

2.1 Acoustic features

Speech recognition systems do not model speech directly at the waveform level; instead signal processing techniques are used to extract the *acoustic features* that are to be modeled by an HMM.[5] A good acoustic feature representation for speech recognition will be compact, without losing much signal information. In practice the acoustic feature representations used in speech recognition do not retain phase information, nor do they aim to retain information about the glottal source which (for many languages) is relatively independent of the linguistic message. Figure 12.1 shows a speech waveform and the corresponding *spectrogram*, a representation that shows the energy of the speech signal at different frequencies. Although a variety of representations are used in speech recognition, perhaps

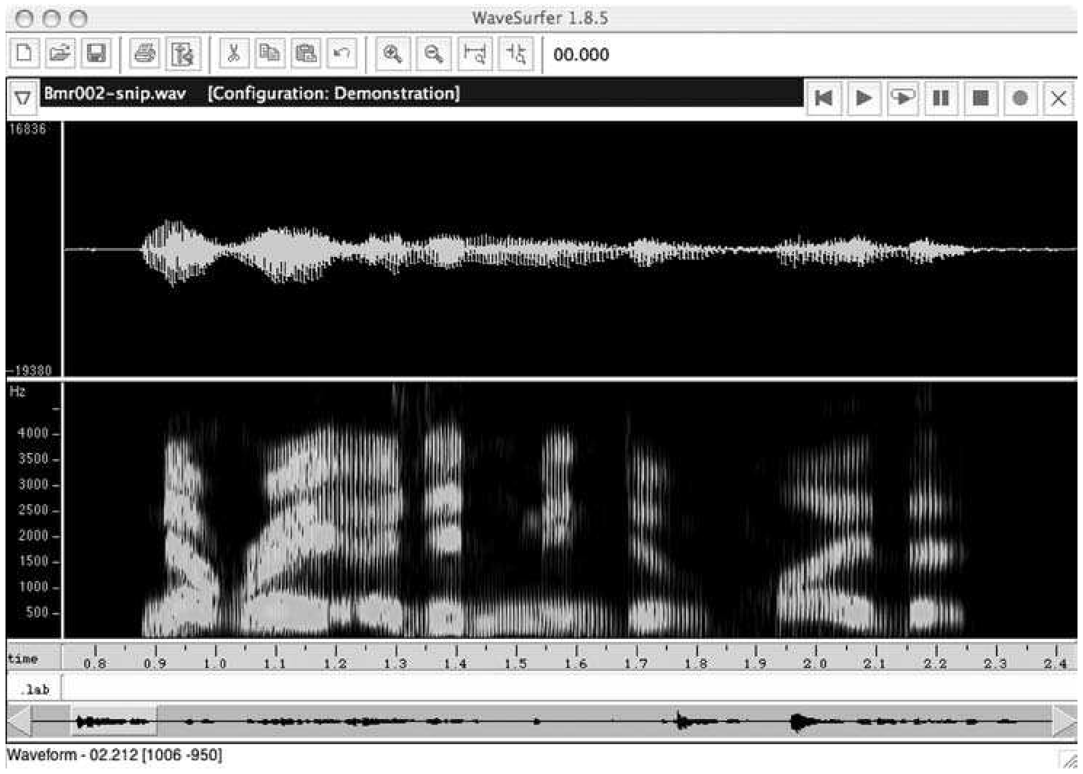


Figure 12.1 Waveform (top) and spectrogram (bottom) of conversational utterance '*no right I didn't mean to imply that*.'

the most widely used are Mel frequency cepstral coefficients (MFCCs) (Davis & Mermelstein 1980). MFCCs are based on the log spectral envelope of the speech signal, transformed to a non-linear frequency scale that roughly corresponds to that observed in the human auditory system. This representation is smoothed and orthogonalized by applying a discrete cosine transform, resulting in a cepstral representation. These acoustic feature vectors are typically computed every 10 ms, using a 25 ms Hamming window within the speech signal. Perceptual linear prediction (PLP) is a frequently used alternative acoustic feature analysis, which includes an auditory-inspired cube-root compression and uses an all-pole model to smooth the spectrum before the cepstral coefficients are computed (Hermansky 1990).

Speech recognition accuracy is substantially improved if the feature vectors are augmented with the first and second temporal derivatives of the acoustic features (sometimes referred to as the deltas and delta-deltas), thus adding some information about the local temporal dynamics of the speech signal to the feature representation (Furui 1986). Adding such temporal information to the acoustic feature vector introduces a direct dependence between successive feature vectors, which is not usually taken account of in acoustic modeling; a mathematically correct treatment of these dependences has an impact on how the acoustic model is normalized – since fewer feature vector sequences will be consistent – and results in an approach that may be viewed as modeling an entire trajectory of feature vectors (Tokuda et al., 2003; Bridle 2004; Zhang & Renals 2006; Zen et al., 2007).

Many state-of-the-art ASR systems use a 39-dimensional feature vector, corresponding to twelve MFCCs (or PLP cepstral coefficients), plus energy, along with their first and second derivatives. These acoustic feature representations have co-evolved with the basic acoustic models used in ASR: HMMs using multivariate Gaussian or Gaussian mixture output probability density functions, discussed in the next section. A particular advantage of cepstral representations compared with spectral representations is the decorrelation of cepstral coefficients, compared with the high correlations observed between neighboring spectral coefficients. Such decorrelations are very well matched with the distributional assumptions that underlie systems based on Gaussians with diagonal covariance matrices.[6] Furthermore, the compact smoothed representations obtained when using MFCC or PLP coefficients results in component multivariate Gaussians of lower dimension than would be obtained if spectral representations were used.

Hermansky et al. (2000) introduced a class of acoustic features that attempt to represent discriminant phonetic information directly. These so-called *tandem* features, derived from phonetic classification systems, estimate phone class posterior probabilities and have proven to be successful when used in conjunction with conventional acoustic features. This is discussed further in Section 5.2.

2.2 HMM/GMM framework

The statistical framework for ASR, introduced in Section 1.2, decomposes a speech recognizer into acoustic modeling and language components (equations 3 and 4). Conceptually, the process works by estimating the probability of a hypothesized sequence of words $\mathbf{W}$ given an acoustic utterance $\mathbf{X}$ using a language model to estimate $P(\mathbf{W})$ (see Chapter 3, STATISTICAL LANGUAGE MODELING) and an acoustic model to estimate $p(\mathbf{X}|\mathbf{W})$.

We can regard the machine that estimates $p(\mathbf{X}|\mathbf{W})$ as a *generative model*, in which the observed acoustic sequence is regarded as being generated by a model of the word sequence. Acoustic models in speech recognition are typically based on hidden Markov models. An HMM is a probabilistic finite state automaton, consisting of a set of states connected by transitions, in which the state sequence is hidden. Instead of observing the state sequence, a sequence of acoustic feature vectors is observed, generated from a pdf attached to each state. A hierarchical approach is used to construct HMMs of word sequences from simpler basic HMMs. The building blocks of an HMM-based speech recognition system are HMMs of 'subword units,' typically phones. A dictionary of pronunciations is used to build word models from subword models, and models of word sequences are constructed by concatenating word models. This approach, illustrated in Figure 12.2, enables information to be shared across word models: the number of distinct HMM states in a system is determined by the size of the set of subword units. In the simplest

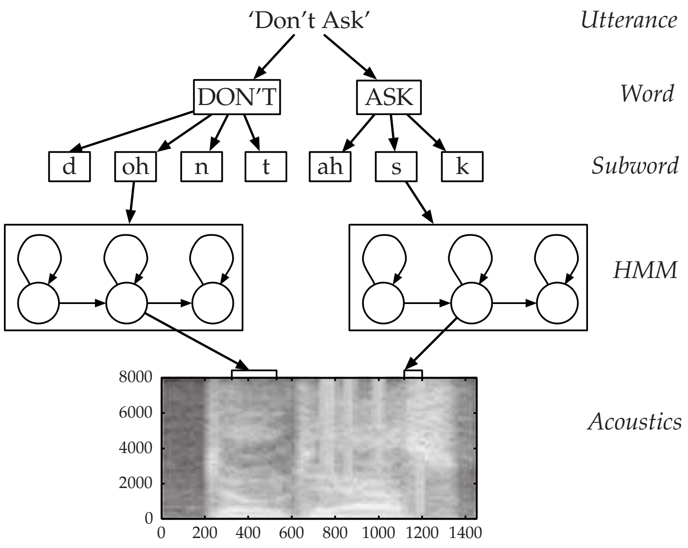
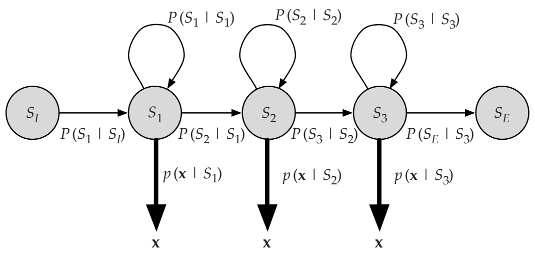


Figure 12.2 HMM-based hierarchical modeling of speech. An utterance model is constructed from a sequence of word models, which are each in turn constructed from subword models.



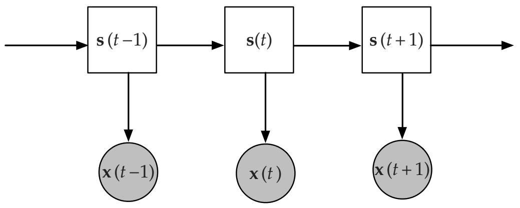


Figure 12.3 Representation of an HMM as a parameterized stochastic finite state automaton (left) and in terms of probabilistic dependences between variables (right).

case, an English speech recognition system might be constructed from a set of 40–60 base phone models with three states each. However, there is a lot of acoustic variability between different observed examples of the same phone. Much of this variability can be accounted for by the context in which a subword appears, and more detailed acoustic models can be achieved using context-dependent subwords, as discussed in Section 2.3. Between-word silence can be represented by special HMMs for silence and, since between-word silence is rare in continuous speech, adding a so-called 'skip' transition to make their existence optional.

An HMM is parameterized by an initial or prior distribution over the states q_i, $P(q_i)$, a state transition distribution $P(q_j \mid q_i)$, and an output pdf for each state $p(\mathbf{x} \mid q_i)$, where $\mathbf{x}$ is an acoustic feature vector. This is illustrated in Figure 12.3 (left) in terms of the model parameters, and in terms of the dependences between the state and acoustic variables in Figure 12.3 (right).

Each state has an output pdf defining the relation between the state and the acoustic vectors. A simple form for the output pdf for state q_i is a d-dimensional Gaussian, parameterized by a mean vector $\boldsymbol{\mu}_i$ and a covariance matrix $\boldsymbol{\Sigma}_i$:

$$(7) \quad p(\mathbf{x}|q_i) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right)$$

For a typical acoustic vector comprising 12th-order MFCCs plus energy, with first and second derivatives, $d = 39$.

Modeling speech using hidden Markov models makes two principal assumptions, illustrated in the graphical model shown in Figure 12.3 (left):

(1) **Markov process**. The state sequence in an HMM is assumed to be a first-order Markov process, in which the probability of the next state transition depends only on the current state: a history of previous states is not necessary.

(2) **Observation independence**. All the information about the previously observed acoustic feature vectors is captured in the current state: the likelihood of generating an acoustic vector is conditionally independent of previous acoustic vectors given the current state.

These assumptions mean that the resultant acoustic models are computationally and mathematically tractable, with parameters that may be estimated from extremely large corpora. It has been frequently argued that these assumptions result in models that are unrealistic, and it is certainly true that a good deal of acoustic modeling research over the past two decades has aimed to address the limitations arising from these assumptions. However, the success of both HMM-based speech synthesis (Yamagishi et al., 2009) and HMM-based speech recognition is evidence that it is perhaps too facile to simply assert that HMMs are an unrealistic model of speech.

Acoustic modeling using HMMs has become the dominant approach due to the existence of recursive algorithms which enable some key computations to be carried out efficiently. These algorithms arrive from the Markov and observation independence assumptions. To determine the overall likelihood of an observation sequence $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T)$ being generated by an HMM, it is necessary to sum over all possible state sequences $q_1 q_2 \ldots q_T$ that could result in the observation sequence X. Rather than enumerating each sequence, it is possible to compute the likelihood recursively, using the *forward algorithm*. The key to this algorithm is the computation of the forward probability $\alpha_t(q_j) = p(\mathbf{x}_1, \ldots, \mathbf{x}_t, q_t = q_j \mid \boldsymbol{\lambda})$, the probability of observing the observation sequence $\mathbf{x}_1 \ldots \mathbf{x}_t$ and being in state q_j at time t. The Markov assumption allows this to be computed recursively using a recursion of the form:

$$(8) \quad \alpha_t(q_j) = \sum_{i=1}^{N} \alpha_{t-1}(q_i) a_{ij} b_j(\mathbf{x}_t)$$

This recursion is illustrated in Figure 12.4.

The decoding problem for HMMs involves finding the state sequence that is most likely to have generated an observation sequence. This may be solved using

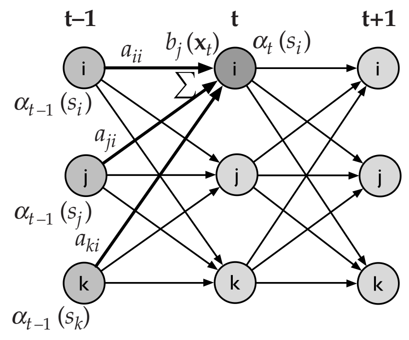


Figure 12.4 Forward recursion to estimate $\alpha_t(q_j) = p(\mathbf{x}_1, \ldots, \mathbf{x}_t, q_t = q_j \mid \boldsymbol{\lambda})$.

a dynamic programming algorithm, often referred to as *Viterbi decoding*, which has a very similar structure to the forward algorithm, with the exception that the summation at each time step is replaced by a max operation, since just the most probable state sequence is required. This is discussed further in Section 3.

By *training* we mean the estimation of the parameters of an HMM: the transition probabilities and the parameters of the output pdf (mean vector and covariance matrix in the case of a Gaussian). The most straightforward criterion to use for parameter estimation is maximum likelihood, in which the parameters are set so as to maximize the likelihood of the model generating the observed training data. Other training criteria may be used, such as maximum a posteriori (MAP) or Bayesian estimation of the posterior distribution, and discriminative training (Section 2.4). Maximum likelihood training can be approximated by considering the most probable state–time alignment, which may be obtained using the Viterbi algorithm. Given such an alignment, maximum likelihood parameter estimation is straightforward: the transition probabilities are estimated from relative frequencies, and the mean and covariance parameters from the sample estimates. However, this approach to parameter estimation considers only the most probable path, whereas the probability mass is in fact factored across all possible paths. Exact maximum likelihood estimation can be achieved using the *forward–backward* or *Baum–Welch* algorithm (Baum 1972), a specialization of the expectation-maximization (EM) algorithm (Dempster et al., 1977). Each step of this iterative algorithm consists of two parts. In the first part (the E-step) a probabilistic state–time alignment is computed, assigning a *state occupation probability* to each state at each time, given the observed data. Then the M-step estimates the parameters by an average weighted by the state occupation probabilities. The EM algorithm can be shown to converge in a local maximum of the likelihood function. The key to the E-step lies in the estimation of the state occupation probability, $\gamma_t(q_j) = P(q_t = q_j \mid \mathbf{X}, \boldsymbol{\lambda})$, the probability of occupying state q_j at time t given the sequence of observations. The state occupation probabilities can also be computed recursively:

$$(9) \quad \gamma_t(q_j) = \frac{1}{\alpha_T(q_E)} \alpha_t(q_j) \beta_t(q_j)$$

where $\alpha_t(q_j)$ is the forward probability for state q_j at time t, $\beta_t(q_j) = p(\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \mathbf{x}_T \mid q_t = q_j, \boldsymbol{\lambda})$ is called the *backward* probability, and $\alpha_T(q_E)$ is a normalization factor (the forward probability for the end state q_E at the end of the observation sequence, time T). The backward probabilities are so called because they may be computed by a recursion that goes backwards in time.

The output pdfs are the most important part of this model, and restricting them to single Gaussians results in a significant limitation on modeling capability. In practice, Gaussian mixture model (GMMs) are used as output pdfs. A GMM is a weighted sum of Gaussians:

$$(10) \quad p(\mathbf{x}|q_i) = \sum_{k}^{K} c_{ik} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})$$

where we have a mixture of K Gaussian components, with mixture weights c_{ik}. Training a GMM is analogous to HMM training: for HMMs the state is a hidden variable, for GMMs the mixture component is a hidden variable. Again the EM algorithm may be employed, with the E-step estimating the *component occupation probabilities*, and the M-step updating the means and covariances using a weighted average.

2.3 *Subword modeling*

As discussed in Section 2.2, and illustrated in Figure 12.2, there is no need to train individual HMMs for each sentence. Instead, the sentence HMMs can be constructed by concatenating *word* HMMs, which in turn may be constructed from *subword* HMMs. This is necessary as training of independent word models is not feasible for most applications: the *Oxford English Dictionary* contains more than 250,000 entries; for morphologically rich languages, such as Finnish or Turkish, there are many more possible words.[7] If we assume that at least 50 samples per word are necessary and take the usually observed average word duration of half a second, the amount of transcribed speech data for English would be approximately 1,700 hours. This calculation is an underestimate, however, as it does not take into account contextual effects, and assumes a uniform distribution over all words. Only very recently have such amounts of annotated data been available, and only for US English. Thus subword modeling seems an attractive alternative: for example the words of British English can be described using a set of about 45 phonemes.

It is possible to write the pronunciation of words in a language with a finite set of symbols, hence one can associate an HMM with each. In the TIMIT database a set of 39 phones is often used: the ARPABET, a phoneme set defined for speech recognition in US English, identifies 40 distinct symbols. By using HMMs for individual phonemes, several important changes are made: a dictionary describing the transitions from words to phonemes has become necessary; the number of units has been drastically reduced; and the length of the units have become more

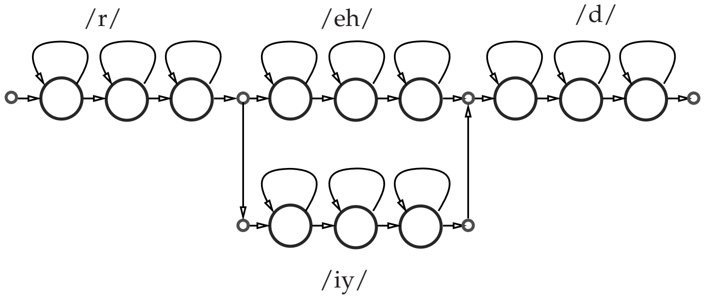


Figure 12.5 Hidden Markov models for phonemes can be concatenated to form models for words. Parallel paths indicate pronunciation variants.

similar. Although the TIMIT corpus was phonetically labeled, this is not a necessary requirement for using models for phonemes, as the phoneme sequence can be inferred from the word sequence using the dictionary and the sentence HMM is constructed by phoneme model concatenation. One issue arises in that the mapping from phonemes to words is not unique, e.g., the word '*read*,' can be pronounced in two ways, /r iy d/ and /r eh d/. However, different pronunciations can be represented by using parallel paths in the sentence HMM (see Figure 12.5). Even more so, different weights can be given to these variants. Pronunciation modeling and/or lexical modeling (e.g., Strik & Cucchiarini 1999) usually focuses on how to best encode pronunciation variability.

Having fewer than 100 *small* HMMs does not allow to capture the large variability of speech, caused by individual variation in articulation and speaker differences (so called intra- and inter-speaker variability). In an attempt to deal with problems arising from acoustic phonetic context, such as coarticulation, many state-of-the-art systems are based on *context-dependent* phone models, often called *triphones*.

Triphone modeling has at its heart the idea of basic sounds that differ depending on their context. In particular, the context here is the immediate neighboring phonemes. Take for example the word '*sound*' and its ARPABET representation /s ow n d/. Instead of having a model for the phoneme /ow/ a model for /ow/ with left context /s/ and right context /n/ is constructed, typically written in the form [s-ow+n]. Taking a typical phoneme inventory size of 45 the number of such triphone models has now risen to 91,125 while retaining the trivial mapping from words to models. In order to avoid the same issues as with word models however, one, can start to declare some triphones as equivalent, e.g., /ow/ may sound very similar to any fricative appearing beforehand.

Most state-of-the-art speech recognition systems make use of clustered triphones or even phone models in a wider context (e.g., quin-/penta-phones, Woodland et al., 1995). A natural grouping approach would be knowledge-based (e.g., Hayamizu et al., 1988), but it has been repeatedly shown that automatic techniques give superior performance. As in many machine learning approaches clustering can operate bottom-up or top-down. The use of agglomerative clustering

(Hwang & Huang 1992) allows robust modeling but, if not all triphones have been observed in the training data, it is unclear how to represent words containing those triphones. Instead classification and regression trees (CARTs) were found to serve well in both cases (Bahl et al., 1991; Young et al., 1994). Here a binary decision tree is grown from a predefined set of questions about the neighboring phonemes (e.g., if the left neighbor is a plosive). When growing the tree, at each point the one question yielding the highest gain in likelihood is chosen. Tree growing is abandoned if the gain in likelihood falls below a predefined threshold. It is standard to cluster HMM states rather than whole models in this way and normally one decision tree per phoneme and per state position is derived. On average decision trees associated with vowels have greater depth than those associated with consonants.

Several weaknesses of the CART approach are known. For example, when clustering it is assumed that states are well represented by a single Gaussian density. Also, hard decisions are made on classes which are better represented as hidden variables (Hain 2001), and the dependency on neighboring phonemes is not the only property that influences sounds. Alternatives include articulatory representations or factorization of graphical models (King et al., 2007).

In English the relationship between sounds and the written forms is looser than in other languages, for example German or Turkish. For these languages it was shown that using the graphemes directly for modeling can have good results (Killer et al., 2003), thus limiting the need for a manually crafted pronunciation lexicon. For tonal languages, questions on tonal context, including the current phoneme, can significantly improve performance (Cao et al., 2000).

2.4 Discriminative training

Learning of HMM parameters can be approached in fundamentally different ways. While generative learning tries to yield good estimates of the probability density of training samples, discriminative learning drives directly at finding those parameters that yield best classification performance. Within the generative family, maximum likelihood (ML) training is the most widely used scheme. The associated objective function is given by

$$\mathcal{F}_{\mathrm{ML}}(\boldsymbol{\theta}) = \log p(\mathbf{X} \mid \boldsymbol{\theta})$$

which implies that the parameters $\boldsymbol{\theta}$ are chosen to maximize the likelihood of the training data. As discussed in Section 2.2, ML training can be efficiently carried out for HMMs, because the Baum–Welch algorithm enables the efficient computation of the state/component occupation probabilities, supplies good asymptotic guarantees, and is relatively easy to code. Although the result of the algorithm is sensitive to the initialization of parameters, practical experience has shown that this is usually of little impact on error rates obtained.

Discriminative training is based on the idea that, given finite training data and a mismatch of the model being trained to the true model, it is better to focus on learning the boundaries between the classes. Discriminative approaches have become increasingly popular and several techniques have been developed over the past two decades. The most important criteria are maximum mutual information (MMI), minimum classification error (MCE), and minimum Bayes risk (MBR). All of these have in common that not only the correct, i.e., reference, sequence is used in training but also all incorrect word sequences. Since summation over all possible incorrect word sequences is not feasible for large-vocabulary speech recognition, the set of *competitors* is usually constrained to those that have significant probability mass compared with the correct sequence. This implies that the output of recognition of the complete training set must be generated, usually a computationally expensive process.

MCE training (Juang & Katagiri 1992) is based on the idea that model parameters only need correction if misrecognition occurs. The objective function to be maximized is based on the ratio between the likelihoods of the correct sequence and of the incorrect ones. The discriminant function is defined as $d(\mathbf{X}|\boldsymbol{\theta}) = g(\mathbf{X}|\boldsymbol{\theta}) - \overline{g}(\mathbf{X}|\boldsymbol{\theta})$ where g denotes the log-likelihood of the correct sentence and $\overline{g}$ is the average likelihood of the competitors $\mathbf{W}_{\text{incorrect}}$:

$$\overline{g}(\mathbf{X}|\boldsymbol{\theta}) = \frac{1}{\eta}\log\left(\frac{1}{M-1}\sum_{\mathbf{W}\in\mathbf{W}_{\text{incorrect}}}\exp\left(\eta\log p(\mathbf{X}|\mathbf{W},\boldsymbol{\theta})\right)\right)$$

Instead of using the above function directly the transition is smoothed with a sigmoid function, which serves as an approximation to a zero/one loss function (i.e., if the value of the discriminant function is larger or smaller than zero). The overall criterion function again is an average over all training samples of the smoothed discriminants. MCE training is mostly used for smaller vocabularies. Its main weakness is the inefficient use of training data since it only considers misrecognized examples. By allowing the smoothed functions to become a step function and the weight η to tend towards ∞ the criterion function can be shown to converge to the misclassification rate and hence has a clear relationship to MBR training (outlined below).

A recently more prominent alternative is MMI training. Although it was introduced at a similar time (discrete densities, Nadas et al., 1988; continuous, Normandin 1991) it was initially feasible only for small-vocabulary tasks or even discrete word recognition, partially because of the substantial computational cost incurred. The gains were substantial and the first large-vocabulary implementation was reported by Valtchev et al. (1997) but the gains were modest in comparison. The MMI objective function is given by

$$(11) \quad \mathcal{F}_{MMI}(\boldsymbol{\theta}) = \mathcal{E}\left\{\frac{p(\mathbf{X}|\mathbf{W},\boldsymbol{\theta})}{p(\mathbf{X}|\boldsymbol{\theta})}\right\} = \mathcal{E}\left\{\frac{p(\mathbf{X}|\mathbf{W},\boldsymbol{\theta})}{\sum_{\mathbf{W}'\in\mathbf{W}_{\text{incorrect}}} p(\mathbf{X}|\mathbf{W}',\boldsymbol{\theta})^{\alpha}P(\mathbf{W}'|\boldsymbol{\theta})^{\beta}}\right\}$$

where $\mathcal{E}\{\cdot\}$ denotes the expectation. This is equivalent to the mutual information between word sequence $\mathbf{W}$ and acoustic features $\mathbf{X}$. Naturally the amount of information transferred is to be maximized. The criterion has an equivalent interpretation as a variant of conditional maximum likelihood (Nadas 1983). Optimization of the above criterion is mostly based on the extended Baum–Welch algorithm (Gopalakrishnan et al., 1989). In contrast to the standard Baum–Welch algorithm, a learning rate factor similar to gradient descent algorithms was introduced. The selection of the optimal learning rate is difficult and was found to be best set such that the variances of the updated model parameters remain sufficiently positive. Nevertheless, usually after only a few iterations, the algorithms tend to diverge.

Povey (2003) introduced two fundamental improvements that allowed not only to stabilize the algorithm, but also to improve performance on large-vocabulary tasks dramatically. Equation 11 shows two so far unexplained factors α and β. These factors allow to scale the contribution of the acoustic and language model components. Such scaling would be of no effect in ML training but equation 11 involves a sum. Scaling the language model scores is important in decoding (see Section 3) with the rationale that acoustic models underestimate the true likelihoods due to independence assumptions. The rationale in MMI training is identical, with even the same scale factor values being used (or the inverse to scale the acoustics down rather than the language model up). Secondly, smoothing of the update equation proved to be important. The ML estimate serves as a much more stable estimate and the addition of fixed amounts of the MMI parameter updates was shown to greatly enhance the stability of the algorithm and thus improve performance. In Povey (2003), so-called I-smoothing adds a predefined weight to the ML estimate in the update equations.

Both MCE and MMI training have interesting interpretations that are intuitive and fit in well with ML training. However, in both cases one assumption is made that is in conflict with the decoding scheme, namely the use of likelihood to assess the correctness of a sentence. This correctness measure then drives the update of parameters. Speech recognizer output, however, is normally assessed with the word error metric which measures performance in the form of insertions, deletions, and substitutions of words (aka minimum edit or Levenshtein distance). For MCE and MMI the word error rate is of no concern, as long as the likelihood of an incorrect sentence is close to the correct, little change is made on the associated HMM parameters.

To alleviate this shortcoming, minimum Bayes risk training was introduced, initially in the form of so-called minimum phone error (MPE) training (Povey & Woodland 2002). Here explicit use is made of the Levenshtein string edit distance between the competing and reference utterances, $L(\mathbf{W}, \mathbf{W}_{ref})$. The criterion function is given by

$$\mathcal{F}_{MBR}(\boldsymbol{\theta}) = \sum_{\mathbf{W} \in \mathbf{W}} L(\mathbf{W}, \mathbf{W}_{ref}) P(\mathbf{W}|\mathbf{X}, \boldsymbol{\theta})$$

The posterior probability of the competing utterances is weighted by the amount of error in the target metric. Optimization of this criterion is more complicated because, as before, the above sum cannot be computed exactly. Whereas in the case of MMI the HMM properties allow reformation at state level, this becomes more complicated here. The error rate is associated with a whole sentence and the error value of a single word is not well defined as the edit distance has no relation to time, i.e., only the errors of the token string are measured. In order to alleviate this problem two steps can be taken. First, a move to smaller units and, second, the measurement of local error led to the MPE criterion function

$$\mathcal{F}_{MPE}(\boldsymbol{\theta}) = \frac{\sum_{\mathbf{R}} P(\mathbf{X}|\mathbf{R},\boldsymbol{\theta})P(\mathbf{R})A(\mathbf{R},\mathbf{R}_{ref})}{\sum_{\mathbf{R}} P(\mathbf{X}|\mathbf{R},\boldsymbol{\theta})P(\mathbf{R})}$$

where $\mathbf{R}$ is a sequence of phonemes and $\mathbf{R}_{ref}$ is the sequence associated with the reference words, and $A(\cdot)$ denotes the count of raw phoneme errors. It can be shown that maximization of this function leads again to the extended Baum–Welch equations and updates similar to those obtained for MMI if the phoneme distance is replaced by a local estimate (Povey 2003) based on an approximation of frame overlap. Gibson (2008) provided a formal proof of local convergence.

Discriminative training allows significant gains in word error rate. On most large-vocabulary tasks, 10–20 percent relative improvement in word error rate is typically found in comparison to ML trained models which in most cases also serve as the starting point for training. The rate of improvement is rather dependent on model set size, and both MMI and MPE training allow much more compact models. Original work on large-scale MMI training postulated that gains increased with an increase in the amount of data (Woodland & Povey 2000), which is very different from the behavior observed for ML training. The computational cost of discriminative training is high and the complex solutions for optimization cause suboptimality in other areas such as speaker adaptation (see Section 2.5). Naturally discriminative training is sensitive to the quality of the reference labels.

2.5 *Speaker adaptation*

Speech signals vary substantially without significant changes in human perception. A single person can vary the signal due to context, mood, or prosody. Despite the range of intra-speaker variations, listeners are easily capable of discerning different speakers. The differences are manifold and include general speech behavior but also the physical characteristics of a person. The vocal tract shape, lungs, and vocal chords, etc., give a distinct characteristic to a person's voice. These characteristics are not as unique as fingerprints but sufficient to yield significant distinction in forensic applications. For the reminder of this section we focus on acoustic adaptation rather than adaptation to linguistic differences, as it is to date the far more common form of adaptation. Nevertheless, lexical and even language models can be adapted to learn speaker-specific characteristics (Strik & Cucchiarini 1999; Bellegarda 2004).

The natural variability of speech sounds themselves (aside from distortions introduced by the environment) is the main source of confusion and cause of errors. For the reasons outlined above it seems logical to separate physical variations from intentional ones. Hence the initial focus of speech recognition research was in the development of systems capable of dealing with a single speaker in order to achieve reasonable performance. In practice, these speaker-dependent (SD) systems are disadvantageous as large amounts of training data have to be collected and transcribed for each speaker, an approach only feasible for select applications.

The HMM-based framework was shown to cope with variability as long as it has been observed in the training data. This allowed the construction of so-called speaker independent (SI) model sets where the HMMs are simply trained on data from multiple speakers. Recognition of utterances from any speaker are then produced with the same models. While this approach is much more practical, the performance of an SI system is substantially inferior to that of an SD system trained on identical amounts of training data. Hence alternatives are required to bridge the gap. Human listeners are capable of adjusting to a new environment or speaker within a few words or sentences. Similarly for ASR, a few sentences can be used to adjust the models for recognition in order to yield better performance either in a second pass of recognition or for further sentences spoken by the target speaker.

Adaptation techniques can be classified in several ways: whether the acoustic models are changed or the extracted features or both (model or feature-based adaption); whether changes to the models are made prior to adaptation (adaptation or normalization); whether the labels used in adaptation can be assumed to be ground truth or with errors (supervised versus unsupervised adaptation). When the features alone are changed, the changes to speech recognition systems are normally small and hence such methods are often preferred. However, it turns out that in most cases changes to the features work best when changing the acoustic models at the same time, referred to as normalization. For most practical applications the ground truth is not known, in which case the adaptation technique must be able to cope with potentially high levels of word error rate. This has a particularly bad effect on discriminative adaptation techniques (Wang & Woodland 2002; Gibson & Hain 2007; Gibson 2008), however one effect usually alleviates that shortcoming: recognition errors are often phonetically similar to the correct word. The phoneme error rate is often lower than the word error rate (Mangu et al., 1999) and hence the correct models may still be chosen.

Since SD performance is assumed to be the best that can be obtained, an initial strategy is to first cluster speaker-specific models into distinct groups. Then the adaptation step would simply consist in finding the most likely group and using the associated model for decoding. Albeit capable of producing SD performance in the limit, it does not make good use of training data, as speaker cluster models are only trained on data of a subgroup and disregarded for the rest. A better approach is provided by the so-called eigenvoices method (Kuhn et al., 1998), where principal component analysis of the parameters of speaker cluster models (the means only are used in the original work) is carried out. Adaptation then is performed by constructing models by weighted combination of these principal components. The weights are found by maximum likelihood optimization on test data.

Another option to bridge the gap between SI and SD model performance is based on the estimation of a prior distribution over the model parameters. The speaker-independent models are used to provide the prior. Maximum a posteriori (MAP) adaptation (Gauvain & Lee 1994) describes the updates of Gaussian (SI) means by data observed in adaptation:

$$\hat{\boldsymbol{\mu}} = \frac{\tau \boldsymbol{\mu}_{\text{prior}} + \sum_{t=1}^{T} \gamma_t \mathbf{x}_t}{\tau + \sum_{t=1}^{T} \gamma_t}$$

The new mean vector is changed from the old $\boldsymbol{\mu}_{\text{prior}}$ with the average observation vector $\mathbf{x}_t$ which is weighted by the posterior probability that the vector has been produced by this Gaussian. The factor τ can be selected to reflect the speed of adaptation and, in practice, iterative application of the above rules shows the best performance (Hain et al., 2005a). While MAP adaptation can yield very good improvements it requires relatively large amounts of adaptation data to ensure that enough Gaussians have actually been observed and changed in the process. Discriminative versions of MAP exist to work with prior models that are already discriminatively trained (Povey et al., 2003).

A more knowledge-driven approach is to target the physical differences between speakers, in particular the vocal tract shape and size, with the latter being the most distinctive feature (e.g., male/female vocal tract sizes differ substantially due to the relative descent of the larynx in males during puberty – see Harries et al., 1998). The standard model of speech production is the source filter model (Fant 1960), with a vocal tract filter represented by linear prediction based on autocorrelation. Here the vocal tract is represented as an ideal acoustic tube with varying length. Change to the length of the tube can be interpreted as a simple shift of the magnitude spectrum with short lengths associated with a more compact spectrum (Cohen et al., 1995; Hain et al., 1999). Vocal tract length normalization (VTLN) implements the so-called warping of the frequency spectrum by shifting the Mel filter banks in MFCC or PLP feature extraction, subject to ensuring proper computation at boundaries (Hain et al., 1999). Furthermore, the optimal warp factor can be found by a search for the one factor that yields the highest likelihood given an HMM set (Hain et al., 1999). However, an SI model set would have been trained on speakers with different vocal tract lengths, increasing the variance of the distributions, an issue that can be avoided by prior training on normalized speaker data. Since the maximum likelihood estimation is used to find the warp factors, models are trained iteratively, interleaving warp factor estimation and model training.

As outlined above, VTLN targets the length of the vocal tract only, allowing the estimation of a single parameter per speaker. This was shown to be equivalent to multiplying the vectors with a matrix of specific structure (Claes et al., 1998). The idea of multiplication with a matrix without constraints leads to one of the most important techniques in speaker adaptation: maximum likelihood linear regression (MLLR) (Leggetter & Woodland 1995). Here the means are adjusted using a linear transform parameterised by a matrix **A** and a bias vector **b**.

$$\hat{\boldsymbol{\mu}}_j = \mathbf{A}_{m(j)}\boldsymbol{\mu}_j + \mathbf{b}_{n(j)}$$

The major difference with MAP adaptation is that the update of the mean of the jth component can use a matrix which is shared across many Gaussian distributions, selected by the index functions $m(j)$and $n(j)$. These functions can be manually set or automatically found, similar to techniques used in triphone state clustering (see Section 2.3). The matrices can be found by maximizing the likelihood of test data using the transformed models. MLLR adaptation is very flexible as the structure of the matrices and vectors can be arbitrarily chosen (e.g., using diagonal transforms only). Variance adaptation is equally possible (Gales & Woodland 1996) and joint optimization with a common transform leads to constrained MLLR, which can be implemented as a feature transform, thus again showing a connection with VTLN. As for VTLN, training on the normalized models also improves: speaker adaptive training can be implemented with constrained MLLR or normal MLLR (Anastasakos et al., 1996).

Speaker adaptation is a rich and extensive topic in automatic speech recognition and the space here is too small to give a full and in-depth account. Many valuable refinements have been made to the fundamental techniques above and some of the newer schemes have not been mentioned. The interested reader will find a good review by Woodland (2001).

3 Search

Given an observed sequence of acoustic feature vectors **X**, and a set of HMMs, what is the most probable sequence of words $\hat{\mathbf{W}}$? This is referred to as the *search* or *decoding* problem, and involves performing the maximization of equations (1–4). Since words are composed of HMM state sequences, we may express this criterion by summing over all state sequences $\mathbf{Q} = q_1, q_2, \ldots, q_n$, noting that the acoustic observation sequence is conditionally independent of the word sequence given the HMM state sequence. If we wish to obtain only the most probable state sequence, then we employ the *Viterbi* criterion, by maximizing over $\mathcal{Q}_{\mathbf{W}}$, the set of all state sequences corresponding to word sequence **W**:

$$(12) \quad \hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}) \max_{\mathbf{Q}\in\mathcal{Q}_{\mathbf{W}}} P(\mathbf{Q} \mid \mathbf{W}) P(\mathbf{X} \mid \mathbf{Q})$$

Thus a decoding algorithm is required to determine $\hat{\mathbf{W}}$ using the above equation and the acoustic and language models.

Solving (12) by naïve exhaustive search of all possible state sequences is of course not feasible. Viterbi decoding (forward dynamic programming) exploits the

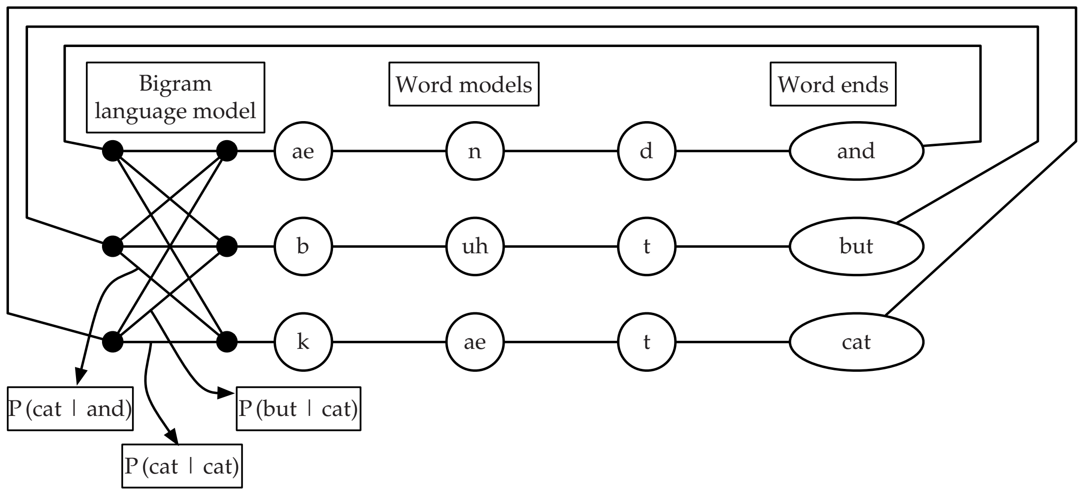


Figure 12.6 Connected word recognition with a bigram language model.

first-order Markov assumption to solve the problem efficiently. Given a state–time lattice, Viterbi decoding carries out a left-to-right time-synchronous processing. This is structurally similar to the forward recursion (Figure 12.4), except that the sum of probabilities of paths entering a state is replaced by a max operation. The Markov assumption is exploited by considering only the most probable path at each point in the state–time lattice: because the history is completely encapsulated by the current state, an extension to a lower probability path at a particular state–time location cannot become more probable than the same extension applied to the most probable path at that state–time location. Thus at each state–time point the single most probable path is retained, and the rest are discarded. The most probable path is the one at the end state at the final time.

To recognize a word sequence, a composite HMM for each word is built (including multiple pronunciations, if necessary) and a global HMM is constructed (Figure 12.6). A bigram language model is easily incorporated; longer span models such as trigrams require a word history to be maintained. As mentioned in Section 2.4, the acoustic model log-likelihoods are often scaled by a factor $0 < \alpha < 1$, which takes into account the underestimate of the likelihood of an observation sequence arising from the conditional independence assumption.

Viterbi decoding is an efficient, exact approach. However, an exact search is not usually possible for a large-vocabulary task since the absence of predefined word boundaries means that any word in the vocabulary may start at each frame. Cross-word context-dependent triphone models and trigram language models add to the size of the search space and overall search complexity. Large-vocabulary decoding must make the problem size more manageable by reducing the size of the search space through pruning unlikely hypotheses, eliminating repeated computations, or simplifying the acoustic or language models. A commonly employed approach to shared computation, which does not include approximation, arises from structuring the lexicon as a prefix pronunciation tree (Gupta et al., 1988;

Bahl et al., 1993) in which common pronunciation prefixes are shared between word models; some extra bookkeeping is required to take account of context-dependent word models (Ravishankar 1996).

The most general approach to reducing the search space – and hence speeding up the decoding – is *beam search*. In beam search (Lowerre & Reddy 1980; Ney & Ortmanns 2000), unlikely search paths are pruned from the search, by removing modes in the time–state trellis whose path probability is more than a factor δ less probable then the best path, defining a beam of width δ. Both the acoustic and language models contribute to pruning in this way. Naïvely, language models are applied at the end of a word, but it is possible to tighten the beam by applying a language model upper bound within the pronunciation tree. Applying beam search means that the decoding process is no longer exact, and hence increases in speed must be traded off against search errors – those errors that arise from incorrectly pruning a hypothesis that would go on to be the most probable. Some form of beam search is used in every large-vocabulary decoder.

Most modern large-vocabulary systems use a multi-pass search architecture (Austin et al., 1991), in which progressively more detailed acoustic and language models are employed – the AMI system, described in Section 4), is a good example of such a system. In multi-pass systems the search space is constrained by constructing word graphs (Ney & Ortmanns 2000) with less detailed models, which are then rescored by more detailed models. Such a system can also be used to combine differently trained models.

In the 1990s most approaches to large-vocabulary decoding constructed the search network dynamically. For a large-vocabulary system, with a trigram language model, constructing the complete network in a static manner seemed out of the question in terms of required memory resources. Several very efficient dynamic search space decoders were designed and implemented (Odell 1995). Although such decoders can be very resource efficient, they result, for instance, in complex software with a tight interaction between the pruning algorithms and data structures. In contrast, searching a static network would offer the ability to decouple search network construction from decoding, and to enable algorithms to optimize the network to be deployed in advance. (Mohri et al., 2000; 2002) have developed an approach to efficient static network construction based on weighted finite state transducer (WFST) theory. In this approach the components of the speech recognition system (acoustic models, pronunciations, language model) may be composed into a single decoding network, which is optimized using determinization and minimization procedures. This can result in a static network of manageable size, although the construction process may be memory-intensive. Such WFST approaches have now been used successfully in several systems. A number of freely available toolkits for WFST manipulation now exist.[8]

An alternative approach to large-vocabulary decoding is based on heuristic search: *stack decoding* (Jelinek 1969; Gopalakrishnan et al., 1995; Renals & Hochberg 1999). In stack decoding (which is essentially an A*-search), a 'stack' (priority queue) of partial hypotheses is constructed, with each hypothesis scored using on the probability of decoding to the current time point, plus an estimate of the remaining score. In this time-asynchronous approach to decoding, a best-first approach is adopted. Since it does not rely on construction of a finite-state network it is well suited to long-span language models and acoustic models.

4 Case Study: The AMI System

Large-vocabulary speech recognition systems have been developed for a number of applications, using the basic acoustic modeling framework outlined in the previous sections. In the research community the most notable examples include read newspaper text, broadcast news, and conversational telephone speech. As discussed in Section 1, read or planned speech has less variability than conversational or spontaneous speech, and this is reflected in the much lower word error rates obtained by automatic speech recognition systems on planned speech tasks. For both planned and spontaneous speech, it has been found consistently that the accuracy of an HMM-based speech recognition system is heavily dependent on the training data. This dependence has two main characteristics. First, accuracy increases as the amount of training data increases (experience has shown the relationship to be logarithmic). Second, the availability of transcribed, in-domain acoustic data for training of acoustic and language models leads to significantly reduced errors. Word error rates can be halved compared with models that were trained on data recorded under different conditions, or from a different task domain.

In this section we consider the construction of a speech recognition system for multiparty meetings. Multiparty meetings, characterized by spontaneous, conversational speech, with speech from different talkers overlapping, form a challenging task for speech recognition. Since much of the work in this area has taken place in the context of the development of interactive environments, the data has been collected using both individual head-mounted microphones and multiple microphones placed on a meeting-room table (typically in some form of array configuration). Here we give a basic description of a system developed for the automatic transcription of meeting speech using head-mounted microphones; the system we outline was developed for the NIST Rich Transcription evaluation in 2006, and some of the final system's complexity has been omitted for clarity.

The meetings domain forms an excellent platform for speech recognition research. Important research issues, necessary to the construction of an accurate system for meeting recognition, include segmentation into utterances by talker, robustness to noise and reverberation, algorithms to exploit multiple microphone recordings, multi-pass decoding strategies to enable the incorporation of more detailed acoustic and language models, and the use of system combination and cross-adaptation strategies to exploit system complementarity. The exploitation of system complementarity has proven to have a significant effect on word error rates. Speech recognition architectures may differ in terms of training data, features, or model topology, and these differences can result in systematically different errors. One can capitalize on such differences in two ways. First, unsupervised cross-adaptation enables the output of one part of the system to adapt another different part. This enables some of the adverse effects caused by unsupervised adaptation to be alleviated. Second, system combination allows the combination of the outputs of several stages of a system, for example by use of majority voting (Fiscus 1997).

Work on meeting transcription has in part been dominated by the fact that the amount of in-domain data is usually relatively small. As for any other spontaneous speech source, the cost of manual transcription is high (manual transcription of meeting data is about 25 times slower than real time). For the system described here, about 100 hours of acoustic training data from meetings were available, which is still modest. Hence most systems make use of adaptation of models from other domains. Stolcke et al. (2004) used a recognition system for conversational telephone speech as the starting point (others have reported that starting from broadcast news systems also works well, e.g., Schultz et al., 2004) and we have followed that strategy. Our experiments with language models for meeting data (Hain et al., 2005b) indicated that the vocabulary is similar to that used for broadcast news, with only a few additional out-of-vocabulary words. Later work has shown that meeting-specific language models can give lower perplexity and word error rate, but the effect is small. Our systems have used a vocabulary of 50,000 words based on the contents of meeting transcriptions augmented with the most frequent words from broadcast news. Pronunciations were based on the UNISYN dictionary (Fitt 2000). The baseline language model for these experiments was a trigram built using the meeting transcripts, a substantial amount of broadcast news data, and most importantly data collected from the internet obtained by queries constructed from *n*-grams in the meeting transcripts (Bulyko et al., 2003; Wan & Hain 2006).

Figure 12.7 presents an overall schematic of the meeting transcription system. The initial three steps pre-process the raw recordings into segmented utterances suitable for processing by the recognizer. A least mean squares echo canceler (Messerschmitt et al., 1989) is applied to alleviate cross-talk in overlapped speech, followed by an automatic segmentation into speech utterances. The segmentation was performed on a per-channel basis and, in addition to using the standard speech recognition features (PLP features in this case), a number of other acoustic features were used including cross-channel normalized energy, signal kurtosis, mean cross-correlation, and maximum normalized cross-correlation (Wrigley et al., 2005; Dines et al., 2006). The segmentation is based on a multilayered perceptron trained on 90 hours of meeting data. This exceptionally large amount of training data for a simple binary classification was necessary to yield good performance. The raw segment output is then smoothed in order to mirror the segmentation used in training of acoustic models.

The initial acoustic models were then trained using features obtained from a 12th-order MF-PLP feature analysis, plus energy. First and second temporal derivatives are estimated and appended to the feature vector, resulting in

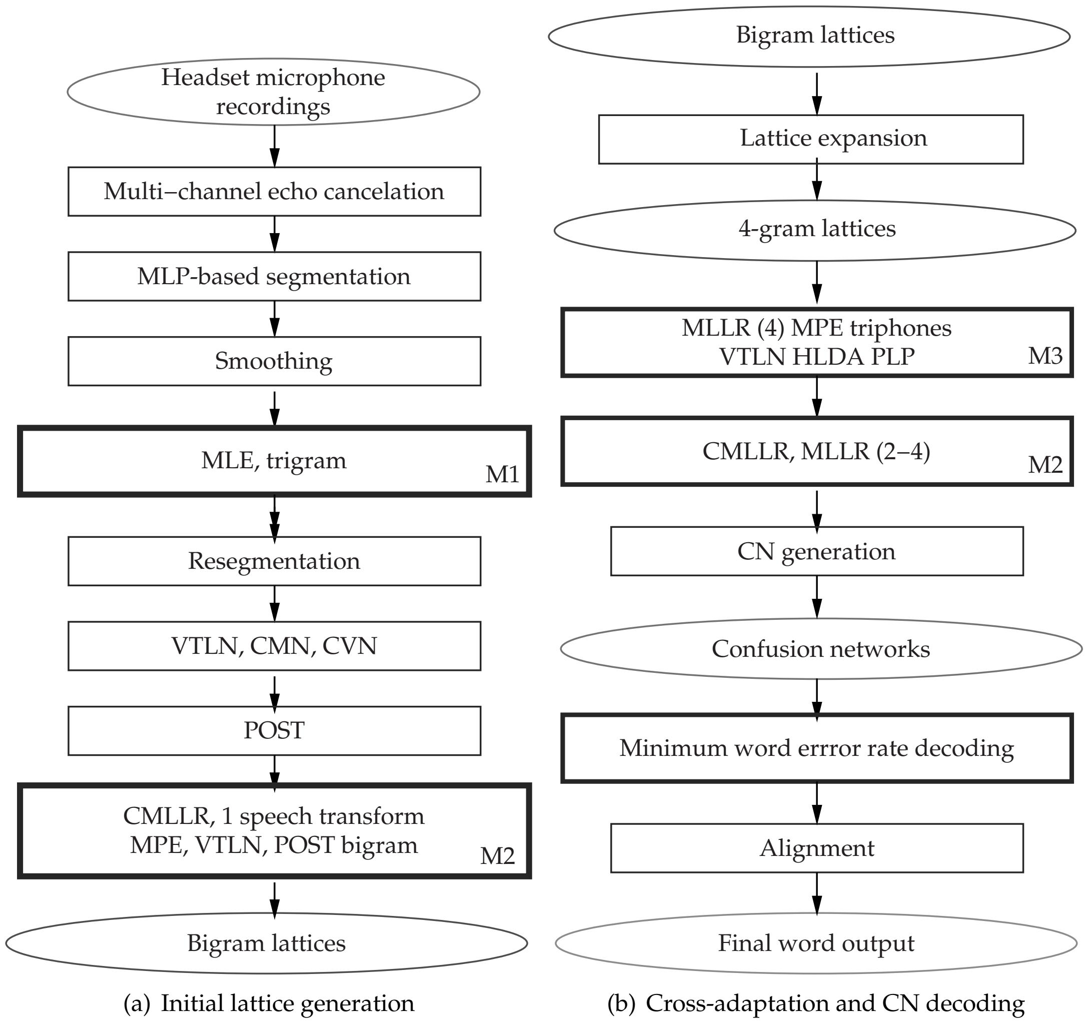

Headset microphone recordings
Multi−channel echo cancelation
MLP-based segmentation
Smoothing
MLE, trigram
M1
Resegmentation
VTLN, CMN, CVN
POST
CMLLR, 1 speech transform MPE, VTLN, POST bigram
M2
Bigram lattices
(a) Initial lattice generation
Bigram lattices
Lattice expansion
4-gram lattices
MLLR (4) MPE triphones VTLN HLDA PLP
M3
CMLLR, MLLR (2−4)
M2
CN generation
Confusion networks
Minimum word errror rate decoding
Alignment
Final word output
(b) Cross-adaptation and CN decoding

Figure 12.7 Block processing diagram showing the AMI 2006 system for meeting transcription (Hain et al., 2006). Square boxes denote processing steps, ellipses representations of the data. M1–M3 denote differently trained model sets.

39-dimensional feature vectors, which are then normalized on a per-channel basis to zero mean and unit variance (cepstral mean and variance normalization).

After these audio preparation stages, 39-dimensional MF-PLP feature vectors were extracted and cepstral mean and variance normalization (CMN/CVN) was performed on a per-channel basis. Then first-pass transcripts are produced, using models trained on 100 hours of meeting data (M1) and a trigram language model. This initial transcript has several uses, including the provision of a rough transcript for estimation of VTLN warp factors, and to allow the data to be resegmented by realigning to the transcripts. This is possible because the acoustic models for recognition are more refined than the models used for segmentation, but naturally segments can only get shorter. This is important as cepstral mean and variance normalization have a significant impact on word error rate and rely on the correct balance of silence in segments.

The next pass of decoding uses different features from those used previously. The standard PLP feature vector is augmented with phone-state posterior probabilities computed using multi-layered perceptrons (further details can be found in Section 5), and both components are normalized using VTLN. Acoustic models are now trained using the MPE criterion (Section 2.4) and further adapted using a single CMLLR transform (Section 2.5). As the system has more passes to follow, bigram lattices are produced at this stage in order to enable lattice rescoring using new acoustic and language models. If a faster system was required, decoding with a trigram as in the first pass could have been chosen here.

The second part of the system, Figure 12.7(b), follows the strategy of using a constrained search space as represented in a lattice to quickly obtain improvements and apply models that could otherwise not be used. First a 4-gram language model (trained on the same data as used previously) is used to expand lattices and produce a new first-best output. This is followed by decoding with two different acoustic model sets for the purpose of cross-adaptation. The first acoustic model set uses standard PLP features only but models are trained by MAP adaptation from 300 hours of conversational data. After adaptation with MLLR, lattices are again produced that are rescored using the same models and features as in the second pass. Finally, lattices are compacted in the form of confusion networks and minimum word error rate decoding is performed. Final alignment is only performed to find correct times for words.

Figure 12.8 shows results for all passes. If the initial automatic segmentation into utterances is replaced by a manual process, then the word error rate is decreased by 2–3 percent absolute. Considerable reductions of word error rates are achieved in each pass. Note that the first pass error rate is almost twice the error rate of the final pass, but the processes of adaptation and normalization in the second pass account for most of the gain. Even though the second pass uses the same acoustic models as the final pass, cross-adaptation still brings an additional 2.4 percent absolute improvement in word error rate.

5 Current Topics

5.1 Robustness

The early commercial successes of speech recognition, for example dictation software, relied on the assumption that the speech to be recognized was spoken in a quiet, non-reverberant environment. However, most speech communication occurs in less constrained environments characterized by multiple acoustic sources, unknown room acoustics, overlapping talkers, and unknown microphones.

A first approach to dealing with additive noise is by using multiple microphones to capture the speech. Microphone array beamforming uses delay-sum

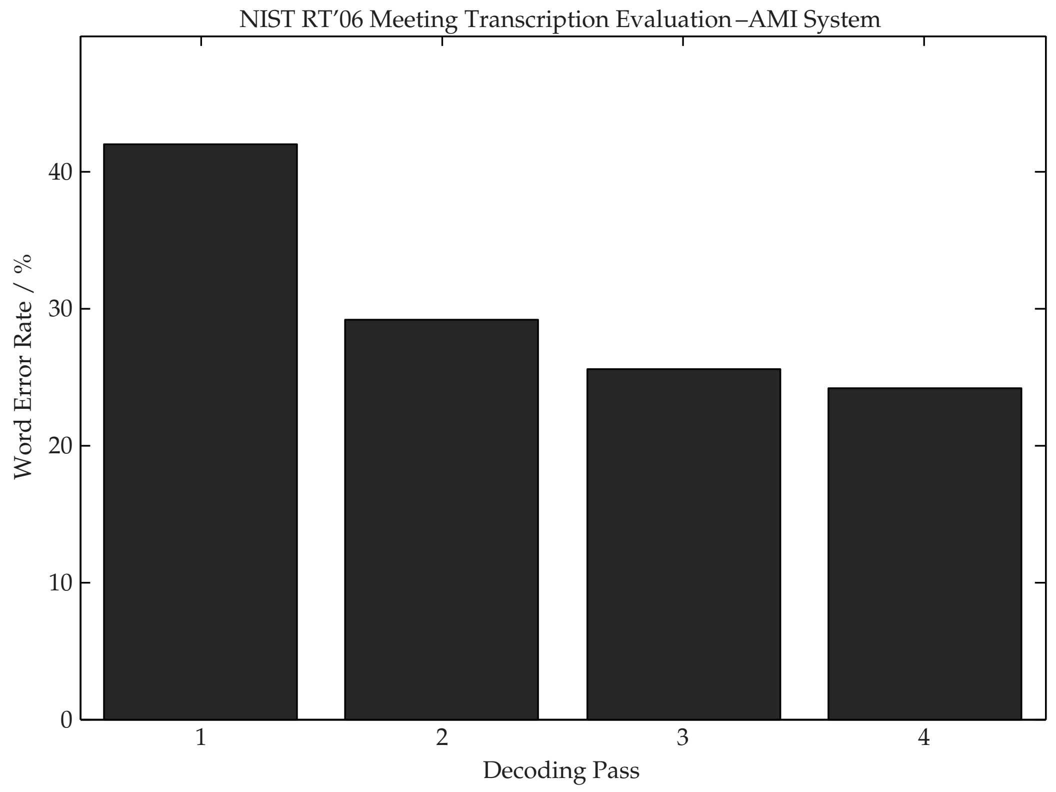


Figure 12.8 Word error rates (%) results in the NIST RT'06 evaluations of the AMI 2006 system on the evaluation test set, for the four decoding passes.

or filter-sum techniques to offer a gain in specific directions, thus enabling competing acoustic sources to be separated based on location (Elko & Meyer 2008). These approaches have been used successfully in meeting transcription systems; using beamformed output of a microphone array will increase the word error rate by about 5 percent in the case of limited overlapping speech (Renals et al., 2008), and by up to 10 percent in more general situations.

Most work in robust speech recognition has focused on the development of models and algorithms for a single audio channel.[9] Increased additive noise will cause the word error rate of a speech recognizer trained in clean, noise-free conditions to increase rapidly. For the Aurora-2 task of continuously spoken digits with differing levels of artificially added noise, the case of clean test data will result in word error rates of less than 0.5 percent using acoustic models trained on clean speech. A very low level of added noise (20 dB SNR – signal-to-noise ratio) results in 10 times more errors (5 percent word error rate) and equal amounts of noise and speech (0 dB SNR) results in a word error rate of about 85 percent (Droppo & Acero 2008). Thus the noise problem in speech recognition is significant indeed.

The reason for the dramatic drop in accuracy is related to acoustic mismatch; the noisy test data is no longer well matched to the models trained on clean speech.

If we knew that the noise would be of a particular type (say car noise) and at a particular level (say 10 dB SNR) then we could artificially create well-matched training data and retrain a set of matched noisy models. However, it is rarely the case that the test noise conditions are known in such detail. In such cases multistyle training can be employed, in which the training data is duplicated and different types of noise added at different SNRs. This can be very effective: in the previous Aurora task, multistyle training decreases the 20 dB SNR word error rate from 5 percent to about 1 percent, and the 0 dB word error rate from 85 percent to 34 percent, while adding about 0.1 percent to the clean speech error rate (Droppo & Acero 2008). Multistyle training is thus very effective, but it is rather computationally expensive: it may be feasible for training a recognizer on digit strings (a task with an 11-word vocabulary); it is much less feasible for conversational speech recognition. Indeed, most of the techniques discussed in what follows have been developed largely on relatively small-vocabulary tasks.

Beyond the brute force approach of multistyle training, there are two main approaches to robustness: feature compensation and model compensation. The aim of feature compensation is to transform the observed noisy speech into a signal that is more closely matched to the clean speech on which the models were trained. It is usually of interest to develop techniques that work in the cepstral feature domain, since speech recognition feature vectors are usually based on PLP or Mel frequency cepstral coefficients. Cepstral mean and variance normalization (CMN/CVN) is a commonly applied technique which involves normalizing the feature vectors, on a component-by-component basis, to zero mean and unit variance. CMN can be interpreted in terms of making the features robust to linear filtering such as that arising from varying type or position of microphones, or characteristics of a telephone channel. For both the Aurora digits task and for large-vocabulary conversational telephone speech recognition, CMN and CVN can reduce word error rates by 2–3 percent absolute (Droppo & Acero 2008; Garau & Renals 2008).

More elaborate forms of feature normalization attempt to directly transform recorded noisy speech to speech that matches the trained models. One technique, designed primarily for stationary noise, is spectral subtraction, in which an estimate is made of the noise (in spectral domain), and subtracted from the noisy speech, to (theoretically) leave just clean speech (Lockwood & Boudy 1992). The subtraction process may be non-linear and dependent on the SNR estimate. This technique requires good segmentation of speech from non-speech in order to estimate the noise. If that is possible the technique works well in practice. The *ETSI Advanced Front End* for noisy speech recognition over cellular phones is based on CMN and spectral subtraction and can reduce errors from 9.9 percent to 6.8 percent on multistyle trained Aurora-2 systems (Macho et al., 2002).

Spectral subtraction may be regarded as a very simple noisy-to-clean mapping, that simply subtracts an estimate of the average noise. More sophisticated approaches are available, for example SPLICE, which is based on the estimation of a parameterized model of the joint density of the clean and noisy speech (Deng et al., 2000). A Gaussian mixture model is typically used, and stereo data containing parallel clean and noisy recordings of the same signal is required for training. Other approaches to feature compensation include missing feature models (Cooke et al., 2001) and uncertainty decoding (Droppo et al., 2002), in which areas of time-frequency space are attributed to speech or noise, and probabilistic enhancement approaches are used to reconstruct the clean speech.

It is also possible to use model-based compensation, in which the detailed acoustic models in the recognizer are used as the basis of the compensation scheme, as opposed to the previous approaches which construct specific feature compensation models. In model-based compensation the clean speech models are combined with a noise model, resulting in a model of noisy speech. This is referred to as parallel model combination (Gales & Young 1996). There are two main technical challenges with parallel model combination. First, noise models that are more complex than a single state result in much greater complexity overall, due to the fact that the speech and noise models are combined as a product. Second, it is assumed that speech and noise are additive in the spectral domain. Since the models are constructed in the cepstral domain, it is necessary to transform the model parameters (Gaussian means and covariances) from the cepstral to the spectral domain. The noisy speech model statistics are then computed in the spectral domain, before transforming the parameters back to the cepstral domain.

5.2 Multiple knowledge sources

It is possible to obtain many different parameterizations of the speech signal and to use different acoustic model formulations. Often different representations and models result in different strengths and weaknesses, leading to systems which make complementary errors. It is possible that word error rates could be reduced if such systems are combined.

In feature combination approaches, multiple different feature vectors are computed at each frame, then combined. Although the most commonly employed acoustic parameterizations such as MFCCs and PLP cepstral coefficients result in low error rates, on average, it has been found that combining them with other representations of the speech signal can lower word error rates. For example, Garau and Renals (2008) combined MFCC and PLP features with features derived from the pitch-adaptive STRAIGHT spectral representation (Kawahara et al., 1999), and Schlueter et al. (2007) combined MFCC and PLP features with gammatone features derived from an auditory-inspired filterbank, each time demonstrating a reduction in word error rate.

The simplest way to combine multiple feature vectors is simply to concatenate them at each frame. This is far from optimal since it can increase the dimensionality quite substantially, as well as result in feature vectors with strong dependences between their elements. The latter effect can cause numerical problems when estimating covariance matrices. To avoid these problems, the feature vectors may be concatenated, then linearly transformed to both reduce dimensionality and decorrelate the components. Although principal component analysis is one way to accomplish this, it has been found that methods based on linear discriminant analysis (LDA) are preferable, since a different transform may be derived for each state. Hunt proposed the use of LDA to improve discrimination between syllables, and in later work used LDA to combine feature streams from an auditory model front end (Hunt & Lefebvre 1988). In LDA a linear transform is found that maximizes the between-class covariance, while minimizing the within-class covariance. LDA makes two assumptions: first, all the classes follow a multivariate Gaussian distribution; second, they share the same within-class covariance matrix. Heteroscedatic LDA (Kumar & Andreou 1998) relaxes the second assumption and may be considered as a generalization of LDA.

As discussed further in Section 5.3, other feature representations have been explored, which have a less direct link to the acoustics. For example, considerable success has been achieved using so-called 'tandem' representations in which acoustic features such as MFCCs are combined with frame-wise estimates of phone posterior probabilities, computed using multi-layered perceptrons (MLPs). An advantage of this approach is that the MLP-based phone probability estimation can be obtained using a large amount of temporal context.

Other levels of combination are possible. Acoustic models may be combined using the combining probability estimates at the frame or at the segment level. Approaches such as ROVER (Fiscus 1997) enable system-level combination in which multiple transcriptions, each produced by a different system, may be combined using a dynamic programming search based on majority voting or on confidence scores. Such approaches have been used to great effect in recent large-scale research systems and are discussed further in Section 5.4.

5.3 *Richer sequence models*

One of the main weaknesses of HMM-based speech recognition is the assumption of conditional independence of speech samples, i.e.,

$$p(\mathbf{x}_t|\mathbf{x}_1 \ldots \mathbf{x}_{t-1}, q_t) \equiv p(\mathbf{x}_t|q_t)$$

where q_t denotes the current state and $\mathbf{x}_t$ denotes the acoustic vector at time t. The conditional independence assumption is incorrect, since it fails to reflect the strong constraints in the speed of movement of articulators, that adjacent frames have high correlation and changes in the spectrum are slow for most sound classes. Some interdependence is of course encoded in the state succession but transition probabilities only exist from one state to the next and are usually found to have a modest influence on performance. This leads to considerable underestimation of true frame likelihoods and two approaches are commonly used to counteract that: the use of differentials (so-called delta features) in the feature vector to account for slope information; and scaling of the language model probabilities in decoding and also training to adjust for dynamic range differences. However, these changes are engineering solutions without a solid theoretical base and also account for part of the shortcoming.

There have been many attempts to address the issue and we cannot present all of those attempted over the years. However, recent interest in better temporal modeling of parameters has increased and considerable improvements have been obtained with some techniques. Sometimes results are difficult to interpret because they include multiple changes to the systems and hence multiple interpretations of the realization are possible. In general all of the techniques are much more computationally elaborate and some are so complex that only rough approximations to the formulae make it possible to realize such structures. Even with modern large-scale computing resources proper implementations are not possible.

Segment models were introduced by Ostendorf and Roukos (1989) and interpreted as an extension to HMM-based modeling (Ostendorf et al., 1996). Instead of modeling single frames, multiple frames can be represented at the same time, for example by describing a moving mean of a Gaussian distribution. Similar to HMMs, one can represent states associated with segments that have variable length. The question on length of the segments and their proper representation is functional; form was (and still is) the topic of investigation, however the techniques have not found entry into large-scale systems, partially due to complexity reasons.

One of the issues pointed out by Tokuda et al. (2000) in the context of HMM-based speech synthesis is the incorrect use of Gaussians for differentials of the static features, noting that an implicit continuity constraint is missing. A formulation that adds this constraint to a standard HMM for recognition is given in Zen et al. (2007). In experiments, significant improvements in word error rates have been observed in some simple tasks, but these improvements have not been observed in more complex tasks (Zhang & Renals 2006).

A much simpler and very effective method has been introduced in the form of the so-called TRAPS features (Sharma et al., 2000), which have been implemented in several large-scale systems in many different applications and modified forms (e.g., the AMI system as presented in Section 4 uses so called LC/RC features as described in Schwarz et al., 2004). The basic idea is to convert long-term information into a single feature vector that is capable of extracting relevant information at the given time. Long-term information can cover up to half a second but most techniques make use of information compression by use of, e.g., a KLT transform on a frequency band basis. The compressed information is then filtered through a multi-layered perceptron that is trained to map features to phoneme state level posterior probabilities. This step is vital as it allows to construct a feature vector that is relevant to the current time without causing information diffusion. Such features can be combined with standard features but recent results seem to suggest that the potential loss in performance is small when they are used on their own. Substantial improvements are obtained on small- and large-scale tasks and gains are often complementary with other techniques.

This technique bears a strong relationship with another technique aimed at augmenting feature vectors. fMPE (Povey et al., 2005), or feature-based MPE training, tries to find a matrix $\mathbf{M}$ such that a new feature vector $\mathbf{y}_t$ is more informative:

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{M}\mathbf{h}_t$$

This is achieved by providing additional information on neighboring frames to the current time in the form of a vector $\mathbf{h}_t$ but, instead of using the features directly, their projection into the model space is performed by using their likelihood for each Gaussian in the acoustic model set, thus providing a link to the models. The matrix is then trained using the MPE type criterion function. Substantial improvements are obtained but training of models and matrices is complex and the application of the matrix in decoding is costly. Hifny and Renals (2009) introduced a related discriminative method: augmented conditional random fields.

Much more advanced approaches try to model long-term dynamics in a more principled form using for example switching linear dynamical systems (Digalakis et al., 1993; Rosti & Gales 2003). Here the fundamental assumption necessary for Viterbi approximation, the assumption that the preceding model does not have an influence on the current model, is normally not correct and hence search paths can not be merged. At this point, for many situations, only small-scale experiments are possible and even then substantial constraints have to be included.

5.4 Large scale

Since the 1990s there has been an intense interest in developing approaches to speech recognition that work well in natural situations. In particular there has been a significant focus on conversational telephone speech, broadcast news, and, more recently, multiparty meetings. Challenges for the recognition of conversational telephone speech include conversational style and the telephone channel. For broadcast news, the speech signals come from a variety of sources and are mixed with other sounds such as music or street noise. The meeting domain adds the challenge of far-field recording and reverberation to conversational speech recognition. Much of this work has been performed in US English, since resources in other languages are usually much more sparse. Lately increased interest in Mandarin and Arabic, as well as 'international English,' has led to extensive resource generation in those languages. Work in different languages brings to the fore important aspects, such as larger vocabularies (over 500,000 words) due to different morphologies, different error metrics, such as character error rate, ambiguity in orthographic and spoken word, or additional sound classes such as Mandarin tones. Remarkably the basic structure in most systems remains identical and changes are mostly made to dictionaries or feature extraction.

The increasing amounts of data as well as the additional acoustic and speech complexity have substantial implications for system building. Segmentation and speaker clustering, optimal for speaker recognition or for playback, is normally not optimal for automatic speech recognition (Stolcke et al., 2004). Multiple microphone sources can affect the best strategies for acoustic modeling and adaptation. Automatic switching between microphones potentially causes substantial errors (Hain et al., 2008). Adapting models to the speaker and to the environment has proven to be extremely important, but the interaction between different model adaptations is complex. In particular, the order of application of techniques is important and may need changing depending on domain or data type.

Dependence on a domain and lack of generalizability are a major challenge. Techniques known to work on one domain do not necessarily work on another domain: this has been observed for VTLN (Kim et al., 2004), for instance. If the in-domain data is sparse, then improved accuracy can be obtained by adaptation from large corpora in related domains. This may require compensation techniques for mismatch between domains; an example of this is the approach of Karafiat et al. (2007) to compensate for different audio bandwidths. Finally, the appropriate evaluation metric may be domain-specific. Word error rate is not the only metric, and optimizing systems for specific applications, such as machine translation (Gales et al., 2007), can lead to significant improvements.

The above illustrates that an almost limitless range of options for system building exists which can serve two fundamentally different purposes: to enhance system performance in one application by finding combinations of components that can enhance performance in another; or to find the components that will yield the perfect result for a particular element of data. Only limited work has been carried out on the latter, but investigations on the AMI RT'07 system (Hain et al., 2007) show that the oracle combination of outputs of various stages of a system can yield 20 percent relative reduction in word error rate.

With the more widespread use of high-level system combination (Fiscus 1997; Evermann & Woodland 2000), recognition systems have become more complex. While attempts were made to describe generic all-purpose system architectures (Evermann & Woodland 2003), experience showed that the search for complementary systems may allow for much simpler structures (Schwartz et al., 2004) where in essence the output of one system is simply used to adapt another one and then the respective outputs are combined. Nevertheless systems that differ by such a margin are difficult to construct. Hence more elaborate schemes, such as in the SRI-ICSI or AMI RT'07 systems (Stolcke et al., 2007; Hain et al., 2007), are developed. In these cases acoustic modeling, segmentation, and data representation is varied to yield complementarity.

The challenges for system development in the future are defined in the list of requirements above. Since the complexity of systems is set to increase rather than decrease, a manual construction of system designs will always be suboptimal. In Hain et al. (2008) initial attempts are reported for automation of system design. However, at this point even the right form to describe the potential combinations efficiently is unknown, let alone a multi-objective dynamic optimization scheme. To find optimal systems, not only does an optimal combination and processing order have to be derived, but ideally the models and techniques are complementary and yield mutually additive gains. Approaches have been made to automate this process (e.g., Breslin & Gales 2006), but much more work is required, in particular in the context of different target metrics.

6 Conclusions

Automatic speech recognition was one of the first areas in which the data-driven, machine learning, statistical modeling approach became standard. Since the 1990s, the basic approach has been developed in several important ways. Detailed models of speech may be constructed from training data, with the level of modeling detail specified by the data. Algorithms to adapt these detailed models to a specific speaker have been developed, even when only a small amount of speaker-specific data is available. Discriminative training methods, which optimize the word error rate directly, have been developed and used successfully. Because of these successful strategies speech recognition is available in commercial products in many forms. Public perception of speech recognition technology, however, ranges widely, from 'solved' to 'hopeless.' The reasons for the mixed acceptance lie in a number of major challenges for speech recognition that are still open today. First, speech recognition systems can only operate in a much more limited set of conditions, compared with people: additive noise, reverberation, and overlapping talkers pose major problems to current systems. Second, the integration of higher-level information is weak and often non-existent, although of obvious use to humans. Third, current models of speech recognition have a rather weak temporal model. The use of richer temporal models has had an inconsistent impact on the word error rate. Finally, systems lack generalizability: they are very dependent on matched training data. Moving a system from one domain to another, without training data resources for the new domain, will result in a greatly increased word error rate.

NOTES

1 This is sometimes referred to as speaker-attributed speech-to-text transcription.

2 An information theoretic measure of the expected number of words which may be expected to continue any word sequence; see Chapter 3, STATISTICAL LANGUAGE MODELING.

3 www.nist.gov/speech/tools

4 http://htk.eng.cam.ac.uk/

5 The autoregressive hidden filter model (Poritz 1982) is an intriguing alternative that performs modeling at the waveform level, and may be viewed as jointly optimizing signal processing and acoustic modeling. However, this approach relies on a linear prediction framework which is less powerful than the approaches employed in current systems.

6 Do not be misled, however; a mixture of diagonal covariance Gaussians is able to model correlations between feature dimensions. But it is a relatively weak way of modeling such correlations.

7 Note that derived forms are counted here as separate words whereas dictionaries such as the *Oxford English Dictionary* only list the base forms as independent entries.

8 WFST software: www.openfst.org/; http://people.csail.mit.edu/ilh/fst/; www.research.att.com/∼fsmtools/fsm/

9 This is a general case, since microphone array beamforming will result in a single audio channel.