

LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

Klaus A J Riederer

Laboratory of Computational Engineering
Helsinki University of Technology (HUT)
P.O. Box 9400, 02015 HUT, Finland
e-mail: Klaus.Riederer@hut.fi, URL: www.lce.hut.fi/~kar

ABSTRACT

Large vocabulary speaker-independent speech recognition systems being capable of recognizing continuous speech based on hidden Markov models are today's standard. This review introduces the fundamentals of speech and the underlying speech recognition problems. The three classical approaches, i.e., the acoustic-phonetic, the statistical (pattern) recognition and the artificial intelligence approach are presented, and the emphasis is put to the most common statistical methods. Finally, the evolution of speech recognizers is illustrated.

1 INTRODUCTION

For humans, speech is the most important manner of communication, to exchange information. The complex speech production/perception process in humans with a machine counterpart is schematically illustrated in Fig. 1. Speech recognition may be defined as a mapping from continuous acoustic signal to a discrete set of symbols. A speech recognizer is a device that automatically transcribes speech into text, thought as a voice-activated "typewriter". Computers execute both speech synthesis and recognition, the prior more successfully than the latter. However, the human flexibility has always been superior to the performance of the machines, to the "artificial intelligence". Humans have been more able to understand synthesized speech than computers to recognize human speech. The reason is the experience of humans with which we can better tolerate the complicated variability in speech signals.

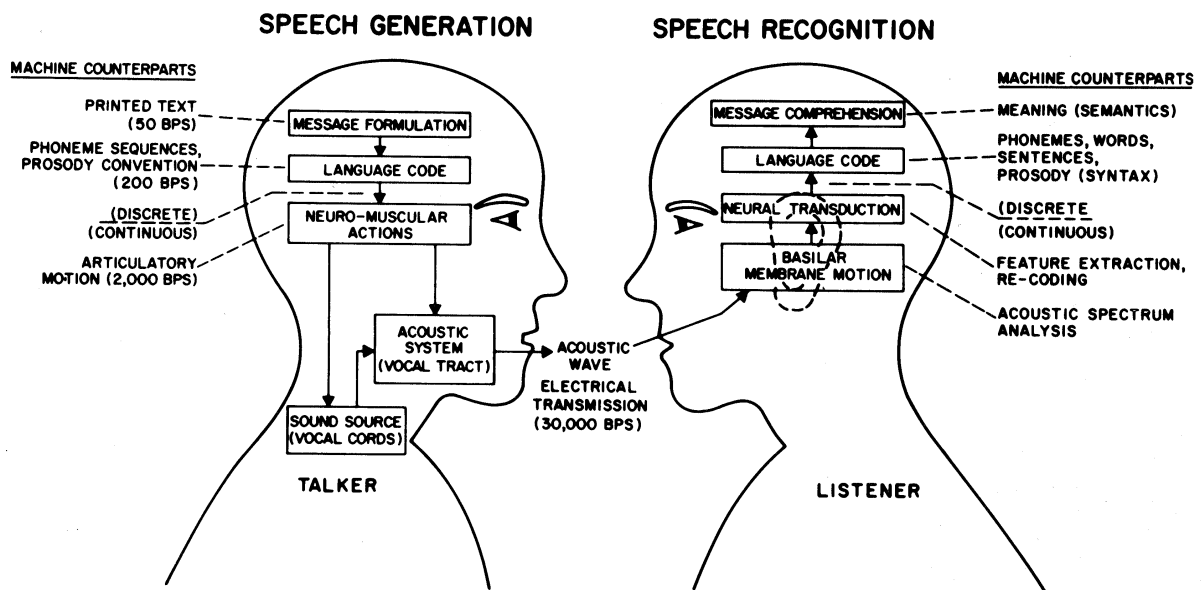


Figure 1. Schematics of speech production/perception process with machine counterparts (Rabiner and Juang 1993). The information rate of raw text is greatly increased at the acoustic signal level.

1.1 FUNDAMENTALS OF SPEECH — ACOUSTIC PHONETICS

Many of the fundamentals of the speech communication process are not well understood. The inner ear functionality is not yet completely revealed, and the brain functions are even a greater mystery; the engineering point of view of a "spectral analyzer" is still leading.

Speech signal is considered as quasi-stationary, i.e., stationary in short time intervals (typically 5-20 ms), during the spectral characteristics are relative constant. Due to the human speech-production process, the *voiced vowels* are *quasi-periodic* with distinct *formants*, but the *consonants* are *transient-like* (formants non-existing) and actually carry more information than the stationary vowels. A spectrogram demonstrating the frequency vs. time (and amplitude) behavior of a speech signal indicates that the *segmentation (labeling)* of the waveform cannot be well defined into regions of silence, unvoiced and voiced signals, see Figs 2 and 5. Speech labeling is in done in the time-domain, where the boundaries of the waveforms are usually not exact, the minor errors in the boundary locations are meaningless for the most applications.

The smallest meaningful unit (linguistically distinct) in speech is called a *phoneme*, which does not have any meaning alone, but it makes possible to discriminate between different words. Speech signals are on the one hand discrete, sequences of linguistically separable units (phonemes), and on the other hand continuous acoustical signals, where the phoneme transitions are mostly relatively smooth. However, the definition of phonemes is somewhat controversial to various linguists; generally it is accepted that in the American English there are 48 phonemes and in the Finnish language 25 different ones. For example, word pair **TALO** — **PALO** presents different phonemes that are not the same as letters. A *phone* signifies the physical sound that is produced when a phoneme is uttered, i.e., the realization of a phoneme. One phoneme can be pronounced in different ways, therefore a phone group containing similar variants of a single phoneme is called an *allophone* (Karjalainen 1999) Human articulatory system has internal inertia, which causes the previous phones to affect the following ones (or vice versa). This "fusion" effect is called *coarticulation*, therefore the articulation of a phoneme depends on its phonemic context.

1.2 GENERAL PROBLEMS IN SPEECH RECOGNITION

Natural speech varies in many aspects. Not only have different speakers different voices, but so is there substantial variation in the voice of a single speaker, due to pronunciation, prosody (stress and intonation of words) or physical state. As a rule, no two uttered sounds are identical. The surroundings are usually noisy, hence the transmitted speech will be degraded. Indeed, an universal and maximally beneficial speech recognizer should be capable of recognizing *continuous speech* from multiple speakers with different accents, speaking styles (including poor articulation), vocabularies, grammatical tendencies and multiple languages in noisy environments. Moreover, the recognizing system should be sufficiently small, low-cost and work in real time. Finally, the system should be

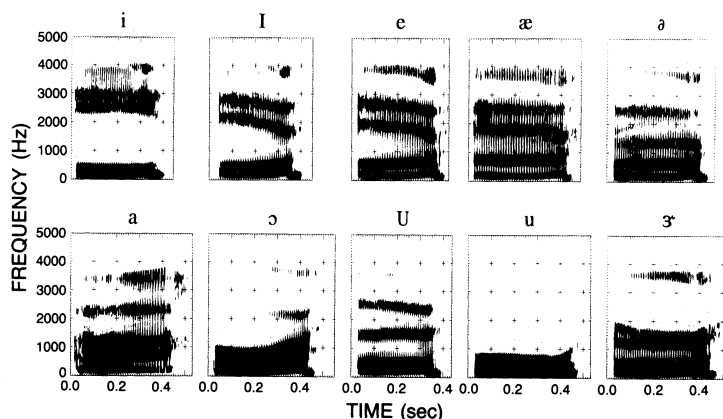


Figure 2. Spectrograms of the vowels sounds (Rabiner and Juang 1993). Horizontal "lines" demonstrate formants that are resonances in the vocal tract; thus they are vowel-specific (Rabiner 1993).

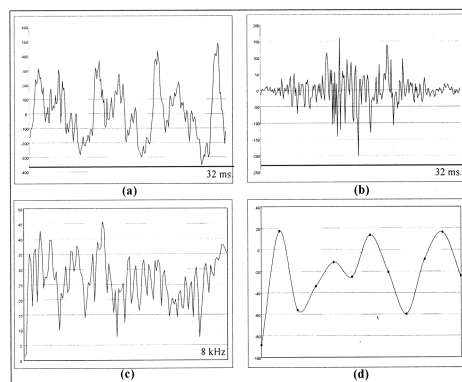


Figure 3. a) speech waveform of a fragment of phoneme /ae/, b) after preemphasis and Hamming windowing, c) power spectrum and d) MFCC (Becchetti and Ricotti 1999)

able to adapt and learn new lexical, syntactic, semantic and pragmatic information as a human can. When considered from this point of view, the field of speech recognition is still in its infancy. Obviously, a speech recognizer is usually limited, including subject-dependency, the size of the vocabulary and the amount of noise in the speech etc.

Based on the previous reasons, the automatic speech recognition (ASR) methods do not follow the authentic human speech recognition that is very efficient and robust. Instead, more general and better-known feature detection and signal processing methods are applied in ASR systems. Moreover, it should not be forgotten that in the speech signal is also embedded information from the identity of the speaker, the language spoken, speech pathologies, and physical and emotional state of the speaker. These personal identity matters are involved in the *speaker recognition/identification* (or *voice identification*) that falls outside the topic of this review.

1.3 SPEECH RECOGNITION APPROACHES

ASR approaches can be divided into different categories, but usually hybrid methods are applied. In general, there are three classical approaches: 1) the *acoustic-phonetic approach*, 2) the *statistical (pattern) recognition* and 3) the *artificial intelligence (AI) approach*. The acoustic-phonetic method is the oldest speech recognition approach originating from the 1950s, the AI approach is the youngest and least known. Statistical methods are by far most commonly applied in modern recognizers. Therefore, this review introduces only briefly the acoustic-phonetic and AI approaches, and concentrates on statistical recognizers.

A speech recognizer consists of “building blocks”, separable components that can be accomplished by various techniques. The components for statistical pattern recognizers are 1) *acoustic processing (front end)*, 2) *acoustic modeling*, 3) *language modeling* and 4) *hypothesis search (linguistic decoding, decoding)*. The front end transforms the acoustic speech signal (sound vibration) into an electric signal and then into symbols that the recognizer will deal with (Jelinek 1997). All ASR approaches utilize a front-end, however, the extracted features depend on the approach in question. Typically, vector quantization, linear prediction coding or cepstrum methods are used in for the feature extraction.

The acoustic-phonetic ASR approach is the most straightforward and thoroughly researched method, whose principles are still used in the AI based recognizers. Therefore, first a brief introduction of the acoustic-phonetic speech recognition is given.

2 ACOUSTIC-PHONETIC SPEECH RECOGNITION

The acoustic-phonetic ASR approach is based on the acoustic phonetics presented previously. The approach assumes that the rules governing the phoneme variability are relatively simple and easily learnable. The first step is the examination of the spectral content of the time-varying speech signal with an analysis system — this *front-end parameterization* is usually done with *linear predictive coding (LPC)*, or (mel-scaled) filter banks with cepstrum analysis, which is presented in more detail in Section 3.2 (see also Fig. 3).

The next step is the feature-detection stage. The LPC coefficients are converted in a parallel fashion to a set of features describing the acoustic properties of the various phonetic units, e.g., nasality (nasal resonance), frication (random excitation), formant locations (frequencies of the first three resonances), voiced/unvoiced classification (periodic or aperiodic excitation) and energy ratios. These features are based on the basic phonetic information, which is gained by spectrogram analysis, i.e., examining the existence and location of formants.

The third step is the heart of the acoustic-phonetic recognizer: the segmentation and labeling phase, in which the system tries to find feature-stable regions and then to label those regions accordingly. The reliability is sensitive, hence various constraints are used as control strategies in order to reduce the search space and improve the accuracy of the segmentation and labeling of the system. In the final stage of *hypothesis search* (or linguistic decoding) the result is a phoneme (or syllable or word) lattice from which the best matching word or sequence of words is lexically determined, see Fig. 5.

The acoustic-phonetic approach has several dilemmas that hinder its functionality as an AVS system. For example, it requires careful a priori knowledge of the phonetic units, even though this information is established a posteriori except for the most simplest cases, e.g., steady vowels. Also, the features are often based on non-optimal

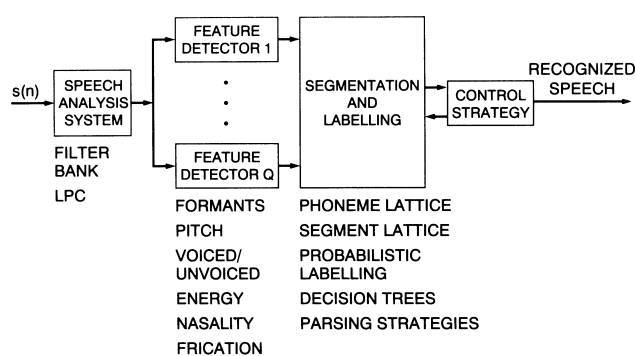


Figure 4. Block diagram of acoustic-phonetic speech recognition system (Rabiner and Juang 1993).

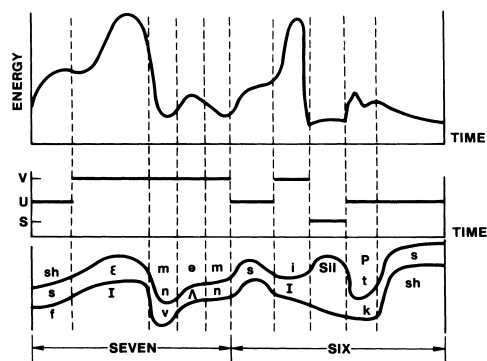


Figure 5. Segmentation and labeling for word sequence "seven-six" (Rabiner and Juang 1993).

ad hoc considerations and the design of the sound classifiers is also non-optimal. Furthermore, there is no well-defined, automatic procedure for tuning the method (i.e., adjusting the decision thresholds) on real, labeled speech. Moreover, there is no standard linguistic way of labeling the training speech. Naturally, these problems need to be solved before the approach can be well utilized in practice.

Due to the above limitations, the acoustic-phonetic speech recognition is no longer considered as the most interesting (single) approach, but it's underlying ideas are still used in the artificial intelligence based recognizers. However, most of the modern recognizers are based on the statistical approach that is presented in detail next.

3 STATISTICAL SPEECH RECOGNITION

The modern large vocabulary recognition (LVR) technology is based on statistical pattern recognition. The underlying principles were pioneered already in the 1970s by speech scientists such Baker, Jelinek and others at IBM, and only a little has changed since. The statistical recognizers are based on hidden Markov models (HMMs), thus the approach is also referred to as the HMM method.

A detailed review of the statistical approach is presented in the following, being mainly based on the paper by Young (1996). Special solutions are shown from the best-known HMM speech recognition system, the HTK Toolkit.

3.1 SYSTEM OVERVIEW

Fig. 6 demonstrates the schematics of an LVR system. The front-end converts an unknown speech waveform into a sequence of acoustic vectors $\mathbf{Y} = \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T$, each representing a short time (ca. 10 ms) speech spectrum of the speech signal. The utterance of a word sequence $W = w_1, w_2, \dots, w_T$ has typically ca. 10 words and a duration of 3 s, yielding a sequence of $T = 300$ acoustic vectors. The basic speech recognition task is to determinate the most probable word sequence \hat{W} , given the observed acoustic signal \mathbf{Y} :

$$\hat{W} = \arg_w \max P(W|\mathbf{Y}) = \arg_w \max \frac{P(W)P(\mathbf{Y}|W)}{P(\mathbf{Y})} . \quad (1)$$

The first term $P(W)$ is the *a priori* probability of observing W independent of the observed signal (sequence) \mathbf{Y} , which is determined by a *language model*. The probability $P(\mathbf{Y}|W)$ is determined by an *acoustic model*.

Fig. 6 shows the computation of the probabilities of a postulated word sequence $W = \text{"This is speech"}$. Each word is converted into a sequence of phones applying a pronouncing dictionary, and for each phone there is a corresponding statistical model, a hidden Markov model (HMM). The sequence of HMMs (representing the postulated utterance) are concatenated to form a single composite model. The probability of that model generating

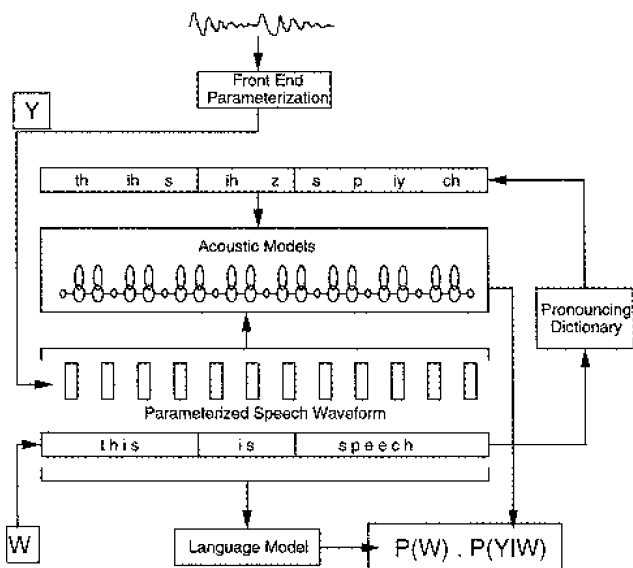


Figure 6. Schematics of an LVR system, showing also the computation of the probabilities of a postulated word sequence $W = \text{"This is speech"}$ (Young 1996).

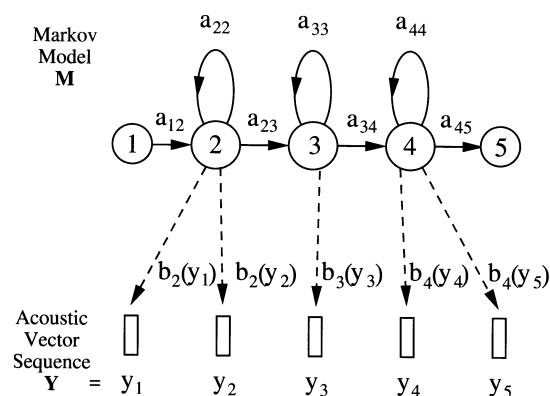


Figure 7. A typical HMM phone model with three emitting states and a simple left-right topology (Young 1996).

the observed Y is calculated, yielding the wanted probability $P(Y|W)$. This *decoding* process may be repeated for all possible word sequences, and the most likely sequence is selected for the recognizer output.

In practice, a (statistical) LVR system requires a number of functional blocks. First, a front-end parameterizes the acoustic speech signal into a compact form for the acoustic model, i.e., HMM. However, the HMMs are always subject to a sparse data problem: the HMM models must precisely represent the distributions of all the contextual cases for each speech sound, and the word predictions have to be based on the preceding history. In addition, the process for searching \hat{W} must occur efficiently in parallel form and improbable hypothesis must be discharged immediately in order to save computation time. Next, these building blocks are presented in more detail.

3.2 FRONT-END PARAMETRIZATION

As stated earlier, a speech signal is considered to be stationary over a short interval. The front-end segments the signal into *blocks* and makes a smooth *spectral estimate* for each block. The (constant) length of the blocks is typically chosen to be 10 ms, and the blocks are overlapped in time to give a longer analysis window of 25 ms (commonly a Hamming window, i.e., a raised cosine). The raw signal is also pre-emphasized, i.e., high frequencies are amplified (ca. 20 dB) in order to compensate for their attenuation because of the mouth directivity. Also other processing is necessary, such as noise suppression and band-pass filtering (usually frequencies limited to 300-3400 Hz) and removal of long silences.

The spectral estimates can be computed via linear prediction or discrete Fourier analysis or cepstrum analysis, and the coefficients, i.e., the final acoustic vectors can be obtained via a number of transformations. The most typical method of modern LVR systems is to use the *mel-frequency cepstral coefficients (MFCCs)*. The processing is mainly done in order to satisfy constraints in the acoustical modeling component.

The Fourier spectrum of each speech block is smoothed by a mel-scale filter-bank that consists of 24 band-pass filters that simulate the human cochlea processing (compare the 24 Bark bands), see Fig. 3. The mel-scale is linear up to 1000 Hz and logarithmic thereafter, creating a so-called *perceptual weighting* to the signal. From the output of the filter-bank a squared logarithm is computed, which discharges the unnecessary phase information and performs a dynamic compression making the feature extraction less sensitive to dynamic variations. This also makes the

estimated speech power spectrum approximately Gaussian. Finally, the inverse DFT is applied to the log filter-bank coefficients, which actually is reduced to a discrete cosine transformation (DCT). DCT compresses the spectral information into lower-order coefficients, and it also decorrelates them allowing simpler statistical modeling. Thus, only diagonal covariance matrixes (instead of full matrixes) can be used in representing the *probability density functions (PDFs)* that are modeled by linear combinations of Gaussian functions. The zero order MFCC coefficient approximates the total energy and it can be omitted as this is already calculated from on the time signal. Generally 9-15 coefficients are used; the HTK recognizer calculates 12 coefficients. Also, perceptually weighted decorrelated LP coefficients give good results.

The acoustic modeling assumes that each acoustic vector is uncorrelated with its neighbors. Due to human articulatory system, this requirement is not well satisfied; there is continuity between consecutive spectral estimates. Second and third order differentials greatly reduce this problem, for example the HTK recognizer fits a linear regression over two preceding and two following vectors resulting the final acoustic vector with 39 components.

3.3 ACOUSTIC MODELING (HIDDEN MARKOV MODELING)

The purpose of the acoustic model is to calculate the likelihood of any vector sequence \mathbf{Y} given word w . For large vocabularies it is essential to decompose a word sequence into basic sounds, i.e., phones. Each individual phone is denoted as an HMM that has a number of states connected by arcs. Fig. 7 illustrates the most typical HMM phone model with *three emitting states* and a simple *left-right topology*. *Composite HMMs* are created by merging the exit state of one HMM with the entry state of another one, and the transition from one HMM to an another occurs from left to right direction. Words are formed from composite HMMs and complete utterances are accomplished from joining words.

An HMM generates *acoustic speech vector sequences* like a finite state machine, at each time t a state j is entered, and a vector \mathbf{y}_t is generated with probability density $b_j(\mathbf{y}_t)$. The transition one speech event to another, i.e., from state i to state j is also probabilistic, and the states have a discrete probability a_{ij} . They constitute a state-transition probability distribution $A = \{a_{ij}\}$, where

$$a_{ij} = P[x(t+1) = j | x(t) = i], \quad 1 \leq i, j \leq N \quad . \quad (2)$$

The observation symbol probability distribution $B = \{b_j(k)\}$, in which

$$b_j(k) = P[\mathbf{o}_t = \mathbf{v}_k | x(t) = j], \quad 1 \leq k \leq M \quad , \quad (3)$$

defines the symbol distribution in state $j, j = 1, 2, \dots, N$. The initial state distribution is $\pi = \{\pi_i\}$, in which

$$\pi_i = P[x(1) = i], \quad 1 \leq i \leq N \quad . \quad (4)$$

Now, the model M is described by its complete parameter set (A, B, π) . The joint probability of a vector sequence \mathbf{Y} and state sequence X , given a model M , is calculated simply as the product of the transition probabilities and the output probabilities \mathbf{o}_i (see Fig. 7):

$$P(\mathbf{Y}, X | M) = a_{12} b_2(\mathbf{o}_1) a_{22} b_2(\mathbf{o}_2) a_{23} b_3(\mathbf{o}_3) \dots = a_{x(0)x(1)} \prod_{t=1}^T b_{x(t)}(\mathbf{y}_t) a_{x(t)x(t+1)} \quad , \quad (5)$$

where the model entry and exit states are constrained to be $x(0)$ and $x(T+1)$, respectively. The observation sequence is denoted as $\mathbf{O} = (\mathbf{o}_1 \mathbf{o}_2 \dots \mathbf{o}_T)$, in which each observation \mathbf{o}_t is one of the symbols from $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_M\}$ and T is the number of observations in the sequence.

In practice, only the vector sequences \mathbf{Y} are observed (known), and the underlying state sequence X is hidden. Therefore, it is called *hidden Markov model*. The desired probability $P(\mathbf{Y} | M)$ is obtained by summing Eq. 5 over all possible state sequences.

Three main problems for HMMs

However, there are three basic issues that need to be solved so that the HMM would be useful in real-world applications (Rabiner and Juang 1993):

1. Given the observation (vector) sequence \mathbf{Y} and a model M , how is the probability $P(\mathbf{Y}|M)$ efficiently computed?
2. Given the observation sequence \mathbf{O} and a model M , how is a corresponding *optimal* (i.e., best explaining the observations) state sequence X chosen?
3. How are the model parameters M adjusted to maximize $P(\mathbf{Y}|M)$?

The first is an evaluation problem; namely, how is the probability computed that the model produced the observed sequence? Or looking the case by another more useful way: scoring how well a given model matches a given observation sequence. As stated just before, the “brute-force” method to solve this problem is to sum the joint probability in Eq. 5 over all possible state sequences. However, this takes on the order of $2TN^T$ calculations, which for even a small case of $N=5$ (states), $T=100$ (observations) requires on the order of $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ computations! Obvious, there needs to be efficient solutions for the calculation of $P(\mathbf{Y}|M)$. The *forward procedure* is based on a lattice (trellis) structure, where there is only N states (nodes at each time slot in the lattice) the amount of which is independent on the length of the observation sequence. This reduces the computation from the order of down to the order of N^2T calculations, i.e., ca. 69 orders of magnitude (!). *Backward* and *Forward-Backward procedures* apply a similar recursive principle and result to the same reduction of calculation.

The second problem of finding the optimal state sequence is more complicated, since are several possible optimality criteria. The most common criteria is to find the states $x(t)$ that are *individually* most likely at each time t . This is done inductively in the *Viterbi algorithm* that is based on dynamic programming methods. It is efficiently implemented in a lattice structure and essential in the decoding phase. Other less ordinary optimality criteria are to solve the state sequence that maximizes the expected number of correct pairs of states $(x(t), x(t+1))$, or triplets of states $(x(t), x(t+1), x(t+2))$ etc.

The third case is the most difficult problem; the model parameters (A, B, π) need to be satisfied to a particular optimization criterion, which then requires training of the HMM. There are two methods applying different likelihood criterion. The *segmental K-means algorithm* uses maximum state optimized likelihood criterion, i.e., the model parameters M are adjusted to maximize $P(\mathbf{O}, X|M)$, where X is now the optimum sequence as given by the Viterbi algorithm. On the other hand, gradient techniques and the significant *Baum-Welch algorithm* use the maximum likelihood criterion, i.e., maximizing locally $P(\mathbf{O}|M)$.

Acoustic discrimination using HMM phone models

Large vocabulary recognizers require efficient acoustic discrimination. Rewriting Eq. 5 in the logarithmic form separates the model parameters a and b :

$$\log P(\mathbf{Y}, X|M) = \sum_{t=0}^T \log a_{x(t)x(t+1)} + \sum_{t=0}^T \log b_{x(t)}(\mathbf{y}_t) . \quad (6)$$

Now, each log probability may be regarded as a score and each transition as cost of moving from one state to another. Every HMM state provides a prototype acoustic vector, and the *log output probability function (OPF)* gives a distance metric allowing the actual acoustic vectors to be compared with the prototype. All spectral variation in real speech is modeled in the critical output probability function. Initial HMM recognizers used discrete OPFs and a *vector quantizer (VQ)*. The acquired acoustic vector was replaced by the index of the closest codebook vector, and OPFs were just look-up tables containing the VQ index probabilities. However, the quantization noise limits the obtained precision. Current systems use parametric continuous-density output distributions that model the acoustic vectors directly; most common is the *multivariate Gaussian mixture*:

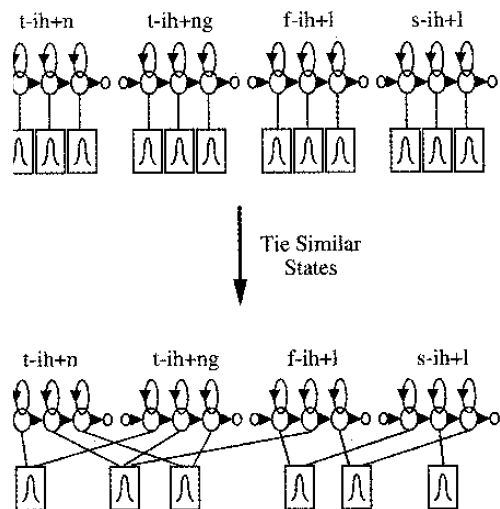


figure 8. State tying (Young 1996).

$$b_j(\mathbf{y}_t) = \sum_{m=1}^M c_{jm} N(\mathbf{y}_t; \mu_{jm}, \Sigma) \quad , \quad (7)$$

where c_{ij} is weight of the mixture component m in state j and $N(\mathbf{y}_t; \mu_{jm}, \Sigma)$ represents a multivariate Gaussian of mean μ and covariance Σ .

In other words, one OPF encloses all possible phones, each of which match the “true” uttered phone according to a certain probability. The “true” word that needs to be recognized is a sequence of the uttered phones. Transitions between different states (each holding different OPFs) constitute possible words (i.e., sequences of the possible phones), each matching the true uttered word with a certain probability.

Contextual effects, such as *coarticulation*, cause large variations to the utterance of words. Therefore, different HMMs need to be trained for each varying context. The most straightforward and common approach to improve phonetic distinction is to use *triphones*, where every phone has an unique HMM model for every distinct pair of left and right neighbors. The so-called *cross-word triphones* model well the contextual effects at word boundaries in fluent speech, but yield more complications in the decoder than the simpler *word-internal-triphones*.

The Gaussian mixture output distributions result to too many trainable parameters in case of triphones. A large-vocabulary triphone recognizer for the English language would need ca. 60 000 triphones, and then the HTK recognizer would have in total 142 200 000 parameters. This problem of excessive amount of parameters and insufficient training data is a key issues in statistical speech recognizers. The recent techniques apply *smoothing based parameter tying*, especially *state-tying* and *phone-based component tying* with continuous-density HMMs give good results. The HTK recognizer employs state tying, where acoustically identical states are coupled together, so that several states share distributions, see Fig. 8. The choice of tying appropriate states is based on *phonetic decision trees*, which use separate binary trees for each phone and state position. The pool is divided into subsets according to phonetic questions (e.g., “is the signal voiced or not?”; see Chapter 2), which maximizes the likelihood of the training data. This gives compact, good-quality state clusters that robustly estimate mixture Gaussian output probability functions. The decision trees can also be used to synthesize an HMM for any possible context, and the HTK recognizer can even use questions spanning ± 2 phones, taking also account word boundaries.

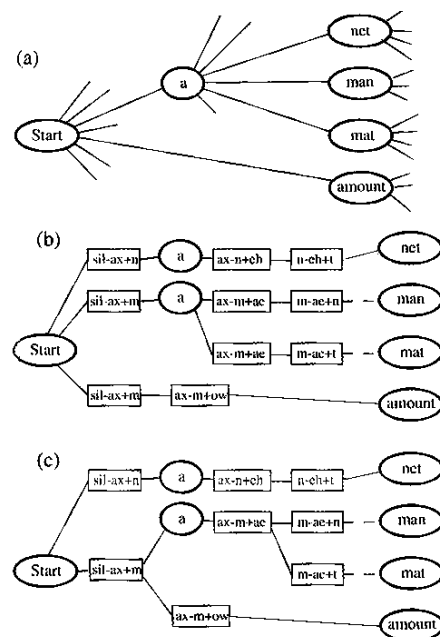


Figure 9. Fragment of a decoder network (Young 1996).

3.4 LANGUAGE MODELING

Language model estimates the probability of some word w_k in an utterance given the preceding words $W_1^{k-1} = w_1 \dots w_{k-1}$. N -grams presume that w_k depends only on the preceding $n-1$ words, i.e.,

$$P(w_k | W_1^{k-1}) = P(w_k | W_{k-n+1}^{k-1}) \quad . \quad (8)$$

N -grams encode simultaneously syntax, semantics and pragmatics and they concentrate on local dependencies. Therefore, they are effective for languages like English in which word order is important and the strongest contextual effects come from near neighbors. N -grams can be computed directly from text data and estimated from simple frequency counts and stored in a look-up table. However, the complexity of the modeling increases logarithmically with the size of the vocabulary V , e.g., for a modest vocabulary of 10000 words there are V^3 possible trigrams ($N=3$). This leads to the sparse training data problem, which can be diminished by careful actions, e.g., with *discounting* (redistributing excess probability mass among the less occurring trigrams) or *backing-off* (trigram probability is replaced by a scaled bigram probability). Other language models have not been successful, hence bigram and trigram language models are most the typical ones in LVR systems. Then again, the previous word-internal-triphones could be defined as unigrams that account for the phonetic variation inside a single word.

3.5 DECODING

The basic recognition problem is to find the sequence of words \hat{W} that maximizes Eq. 1. The *decoder* solves this search dilemma, and there are two main approaches: *depth-first* and *breadth-first*. In the former type, the best hypothesis is pursued (in series) until the end of the speech is reached, examples are *stack-decoders*, *A*-decoders* and *envelope search*.

In the breadth-first approaches, all hypothesis are pursued in parallel. Because LVR systems are complex and pruning of the search space is required, a so-called *beam search* process is normally applied. The decoding exploits Bellman's optimality principle and is referred as *Viterbi decoding*. The HTK toolkit decoder applies beam search and Viterbi decoding.

The branching tree of HMM-state nodes are connected by state transitions and word-end nodes are connected by word transitions, see Fig. 9. Any path from the start node to an arbitrary point in the tree is denoted by a movable *token* placed in the node at the end of the path. The *score* of the token is the total log probability up to that point, and the *history* of the token records the sequence of word-end nodes that the token has passed through. Any path can be extended by moving the token from its current node to an adjoining node and updating its score accordingly. Hence, the search problem is now converted into a basic token-passing algorithm that is guaranteed to find the best possible path, but due to computing limitations, it needs to be pruned. For every time frame, the best score in any token is noted and any token that lies more than a beam-width below this best score is destroyed. This means that only a fragment of the branching tree described in Fig. 9 is ever needed at one time. In order to make this efficient, the tokens need to be pruned as soon as possible. The tree-structure introduces another problem of raising the active number of tokens, since the identity of a new word is unknown until its end node is reached. The HTK decoder solves this by associating a list of all possible current words with every token. Tokens receive their score equal to most likely word in the current list. The list is updated on every model transition, and the tokens are pruned accordingly.

The dynamic performance in the HTK decoder accomplishes a system capable of exploiting complex language models and HMM phone models depending on both the previous and succeeding acoustic context, such as coarticulation. Moreover, it can do this in a *single pass*, in contrast to most other Viterbi-systems that use multiple passes.

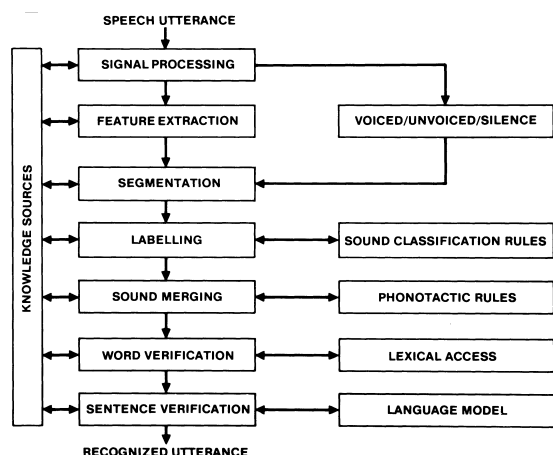


Figure 10. The standard “bottom-up” AI based speech recognizer (Rabiner and Juang 1993).

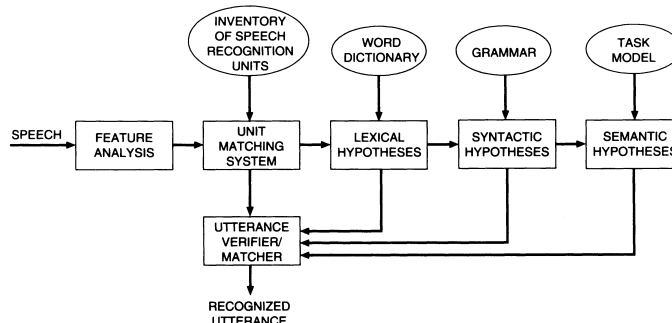


Figure 11. The standard “top-down” AI based speech recognizer (Rabiner and Juang 1993).

4 ARTIFICIAL INTELLIGENCE APPROACHES TO SPEECH RECOGNITION

The artificial intelligence (AI) approach is a hybrid of the acoustic-phonetic and statistical recognition methods. It mimics the human intelligence in visualizing, analyzing and decision making progress on the measured acoustic features. The main idea of AI is to collect and employ knowledge from a number of sources in order to solve the problem in question. The *knowledge sources* (KS) are wide-ranging from the fields of *acoustic*, *lexical*, *syntactic*, *semantic* and *pragmatic knowledge* (Rabiner and Juang 1993).

The KS can be incorporated in many ways to a speech recognizer. In the standard “*bottom-up*” (serial) processor the lowest-level processes (e.g., feature extraction, phonetic decoding) precede higher-level processes (lexical decoding, language model) sequentially so that each state is minimally constrained (see Fig. 10). The alternate “*top-down*” (parallel) processor integrates the word hypothesis matching, lexical decoding and syntactic analyses blocks into a consistent framework, similarly as in the LVS recognizer systems (see Fig. 11).

Another possibility is the so-called *blackboard approach*, in which all KS are regarded independent. A *hypothesis-and-test paradigm* is applied as the main communication medium among KSs that are data-driven, based on the patterns (matching the KS templates) on the blackboard. The system operates asynchronously, and assigned cost and utility considerations are distributed across all levels. The blackboard approach was extensively studied at Carnegie Mellon University in the 1970s, and it has been further researched for dialogue-based expert systems especially at Massachusetts Institute of Technology.

Most important techniques in this approach are the use of an *expert system* for segmentation and labeling of the acoustic signal, learning and adaptation over time, the use of *artificial neural networks* (ANNs) for distinction between similar sound classes and learning the relations between all known inputs and phonetic data. The neural networks could represent a separate structural approach to speech recognition or regarded as an implementation architecture possibly incorporated in any of the three classical speech recognition approaches. The applications of artificial neural networks (ANN) to speech recognition are the youngest and least well understood of the recognition technologies. The oldest examples are applications combining artificial neural networks with conventional technologies originate from the late 1980s applying, e.g., vowel, word and digit recognition using *multi-layer perceptrons* (MLPs), *learning vector quantization* (LVQ) and *time delay neural networks* (TDNNs). The modern methods are *hybrids of ANNs and HMMs* applying, e.g., *recurrent neural networks* (RNNs), *self-organizing maps* (SOMs) and *mixtures of experts* (Rabiner and Juang 1993, Deller et. Al 1993).

5 EVOLUTION OF LVR SYSTEMS

The large vocabulary speech recognizers have been improved enormously since their introduction; 25 years ago they used *isolated word (IW)* recognition with small vocabularies (see Table 1). There are two classes of speech recognition technology: *speaker-dependent*, where the user has to train the system to recognize the user's voice, and *speaker independent*. Speaker independence allows fast and easy implementation since their no prior knowledge of the user is needed. However, often *speaker adaptation* (based on, e.g., MLLR technique) is desired for better performance (and e.g., improved noise robustness) (Koppinen 2000). In any case, all well performing recognition systems are based on hidden Markov models.

Until 1997, large vocabulary systems were limited to discrete speech recognizers that required users to pause between each word. In 1997, large vocabulary products with *continuous speech (CS)* recognition engines were released. Now, companies such as Dragon, IBM, Philips and Apple have commercial CS speaker-independent and (trainable) recognizers with 35000-300000 word vocabularies. *Context dependent (CD)* vocabularies are used for better recognition rates; there are clearly more products available for PCs than other computer platforms (Jan 2000).

The HTK Toolkit is the most sophisticated and widely used large-scale system for speech research, and most recognizers are taught with the HTK. According to the US Advanced Research Project Agency (ARPA) November 1994 evaluation, the recognizer from the CUED HTK group yielded best results. It was able to transcribe continuous speech (CS) speaker-independently with average word-error rates between 5-10%, with 2-3 min speaker adaptation the values dropped well below 5% for most speakers (Young 1996).

In the last few years no real break-throughs have occurred in the speech recognition technology, the results are getting better by just improving the old techniques and larger test data sets.

6 REFERENCES

- Becchetti C. and Ricotti L. P., 1999. *Speech Recognition. Theory and C++ Implementation*. John Wiley & Sons, New York, USA.
- Deller J., Proakis J. and Hansen J., 1993. *Discrete Time Processing of Speech Signals*. MacMillan Publishing Company, USA.
- Jan, 2000. Speech Recognition Systems. The Job Accommodation Network, 12.5.2000. <URL: <http://www.jan.wvu.edu/media/Speechrec.html>>
- Jelinek F., 1997. *Statistical methods for speech recognition*. MIT Press, Massachusetts, USA, 283 p.0
- Karjalainen M., 1999. *Kommunikaatioakustiikka*. Teknillinen korkeakoulu, Akustiikan ja äänenkäsittelytekniikan laboratorio, raportti 51, Espoo, Finland, 237 p.
- Koppinen K., 2000. Personal communication. Tampere University of Finland.
- Lee K.F., 1989. *Automatic Speech Recognition-The Development of the SPHINX System*. MIT Press, USA.
- Rabiner L. and Juang B-H., 1993. *Fundamental of Speech Recognition*. Prentice Hall Signal Processing Series, New Jersey, USA, 507 p.
- Young S., 1996. A Review of Large-vocabulary Continuous-speech recognition. *IEEE Signal Processing Magazine*, Sept. 1996, pp. 45-57.

Table 1. Performance of different speech recognition systems in 1975-1989 (Lee 1989).

System	Developer	Year	Approach	Accuracy (%)	Vocabulary (words)
Itakura	NTT	1975	DTW	97.3	200
Dragon	CMU	1975	KS, statistical	84	194
Hearsay	CMU	1975	Blackboard	87	1011
Harpy	CMU	1976	Dragon & Hearsay	97	1011
Bell Labs	Bell Labs '82	1982	IW	91	129
Feature	CMU	1983	Feature matching	>90	(Isolated English letters)
Tangora	IBM	1985	IW, CS	>97	5000
Byblos	BBN	1988	CD phonemes, adaptive	93	997
Bell Labs	Bell Labs '88	1988	HMM (connected digit)	97.1	
Sphinx	CMU	1989	HMM	96.2	997