4. SPEAKER IDENTIFICATION SYSTEMS

4.1 Overview

This Chapter describes the basic principles of speaker identification systems. The steps of speaker recognition, speaker pruning, text dependent and text independent speech recognition systems are described briefly.

4.2 Speaker Recognition

As human beings, we are able to recognize someone just by hearing him or her talk. Usually, a few seconds of speech are sufficient to identify a familiar voice. The idea to teach computers how to recognize humans by the sound of their voices is quite evident, as there are several fruitful applications of this task.

If the system is provided with the information that all possible test utterances belong to one of the speakers that have been learned by the system, we have a "closed set" of training speakers. If a test utterance may be originating by a person that has not been shown to the system before, we speak of an "open set" of speakers. The system should be able to make a rejection in this case.

Speaker recognition [14] is basically divided into two-classification: speaker recognition and speaker identification and it is the method of automatically identify who is speaking on the basis of individual information integrated in speech waves. Speaker recognition is widely applicable in use of speaker’s voice to verify their identity and control access to services. Adding the open set identification case in which a reference model for an unknown speaker may not exist can also modify above formation of speaker identification and verification system.

Speaker recognition can also be divided into two methods, text-dependent and text-independent methods. In text dependent method, the speakers’ words or sentences have the same text for both training and recognition trials. Whereas the text independent recognition does not rely on a specific text being speak. Formerly text dependent methods were widely in application, but later text independent is in use. Both text dependent and text independent methods share a problem however.

4.3 Speaker Identification

Speaker identification is an easy task for human auditory system. On the man machine interface perspective it is still a difficult problem to solve, because we cannot generate such specific feature sets to ease and to make more robust the identification of speakers by computers. Also, popularity of the topic has increased parallel to the increasing demand of interactive services over the telephone and the Internet, such as telephone and Internet banking which require high levels of security.

Speaker identification is a difficult task [15], and the task has several different approaches. The state of the art for speaker identification techniques include dynamic time warped (DTW) template matching, Hidden Markov Modeling (HMM), and codebook schemes based on vector quantization (VQ). In this thesis, the vector quantization approach will be used, due to ease of implementation and high accuracy.

Speaker identification has also been applied to the verification problem, where the simple rank based verification method was proposed. For the unknown speaker's voice sample, K nearest speakers are searched from the database. If the claimed speaker is among the K best speakers, the speaker is accepted and otherwise rejected. Similar verification strategy is also used in.

Speaker identification and adaptation have potentially more applications than verification, which is mostly limited to security systems. However, the verification problem is still much more studied, which might be due to:

(1) Lack of applications concepts for the identification problem,

(2) Increase in the expected error with growing population size,

(3) Very high computational cost.

Regarding the identification accuracy, it is not always necessary to know the exact speaker identity but the *speaker class* of the current speaker is sufficient (speaker adaptation). However, this has to be performed in real-time.

4.3.1 VQ based speaker identification

The components of a typical VQ-based speaker identification system are shown in Figure 4.1. *Feature extraction* transforms the raw signal into a sequence of 10- to 20 dimensional feature vectors with the rate of 70-100 frames per second. Commonly used features include *mel-cepstrum* (MFCC) and *LPC-cepstrum* (LPCC). They measure short-term spectral envelope, which correlates with the physiology of the vocal tract.

In the training phase, a speaker model is created by clustering the training feature vectors into disjoint groups by a clustering algorithm. The *LBG algorithm* is widely used due to its efficiency and simple implementation. However, other clustering methods can also be considered. The result of clustering is a set of *M* vectors, *C* = *{C1, C2,.....CM},* called a *codebook* of the speaker.

In the identification phase, unknown speaker’s feature vectors are matched with the models stored in the system database. A match score is assigned to every speaker. Finally, a 1-out-of-*N* decision is made. In a closed-set system this consists of selecting the speaker that yields the smallest distortion. The match score between the unknown speaker’s feature vectors $X = \{x_1 \dots \dots x_T\}$ and a given codebook $C = \{c_1 \dots \dots . . c_M\}$ is computed as the average quantization distortion

$$D_{avg}(X, C) = \frac{1}{T}\sum_{i=1}^{T} e(x_i, C)$$

4.3.2 Real time speaker identification

The proposed system architecture is depicted in Figure 4.2. The input stream is processed in short buffers. The audio data in the buffer divided into frames, which are then passed through a simple energy-based silence detector in order to drop out non information bearing frames.

For the remaining frames, feature extraction is performed. The feature vectors are pre-quantized to a smaller number of vectors, which are compared against active speakers in the database. After the match scores for each speaker have been obtained, a number of speakers are pruned out so that they are not included anymore in the matching on the next iteration. The process is repeated until there is no more input data, or there is only one speaker left in the list of active speakers.

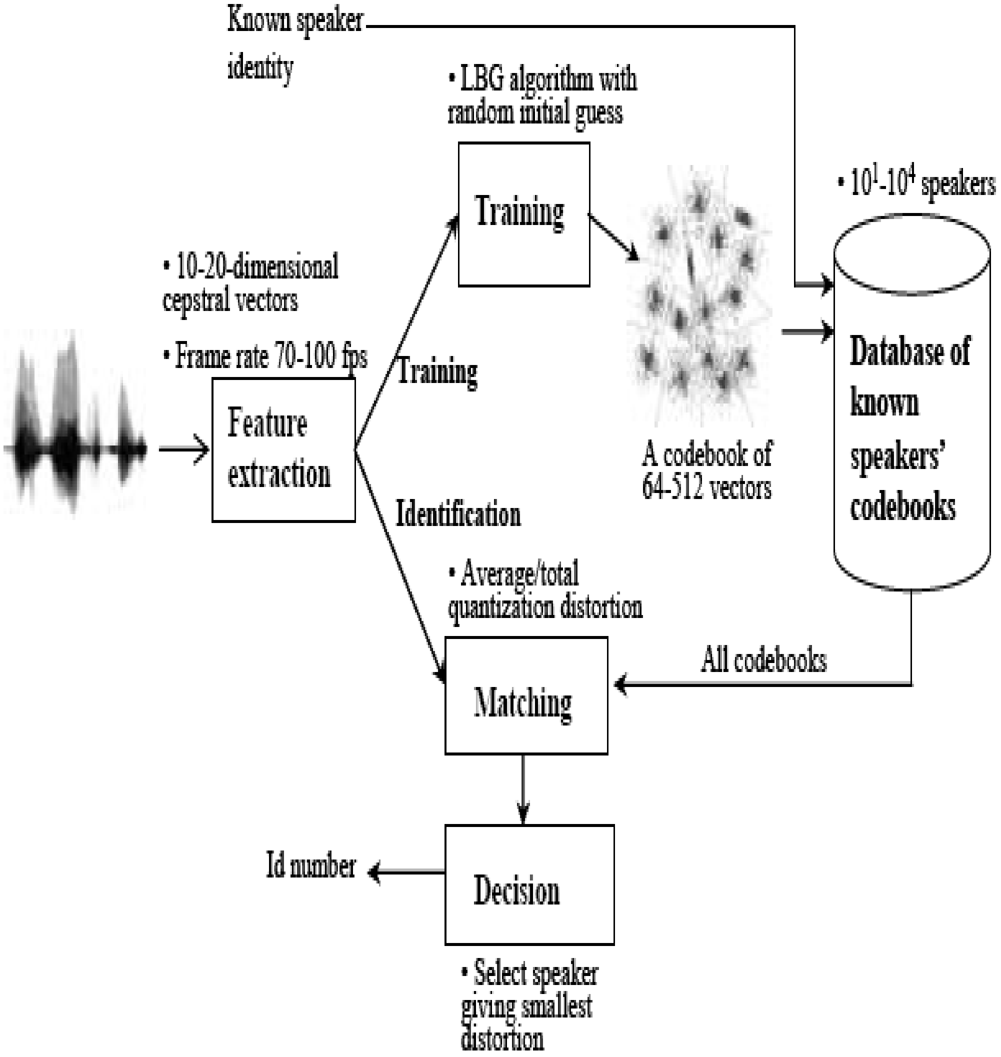

Known speaker identity
• LBG algorithm with random initial guess
Training
• 10^1-10^4 speakers
• 10-20-dimensional cepstral vectors
• Frame rate 70-100 fps
Training
Database of known speakers' codebooks
Feature extraction
A codebook of 64-512 vectors
Identification
• Average/total quantization distortion
All codebooks
Matching
Id number
Decision
• Select speaker giving smallest distortion

Figure 4.1 Typical VQ-based closed set speaker identification system.[4][10]

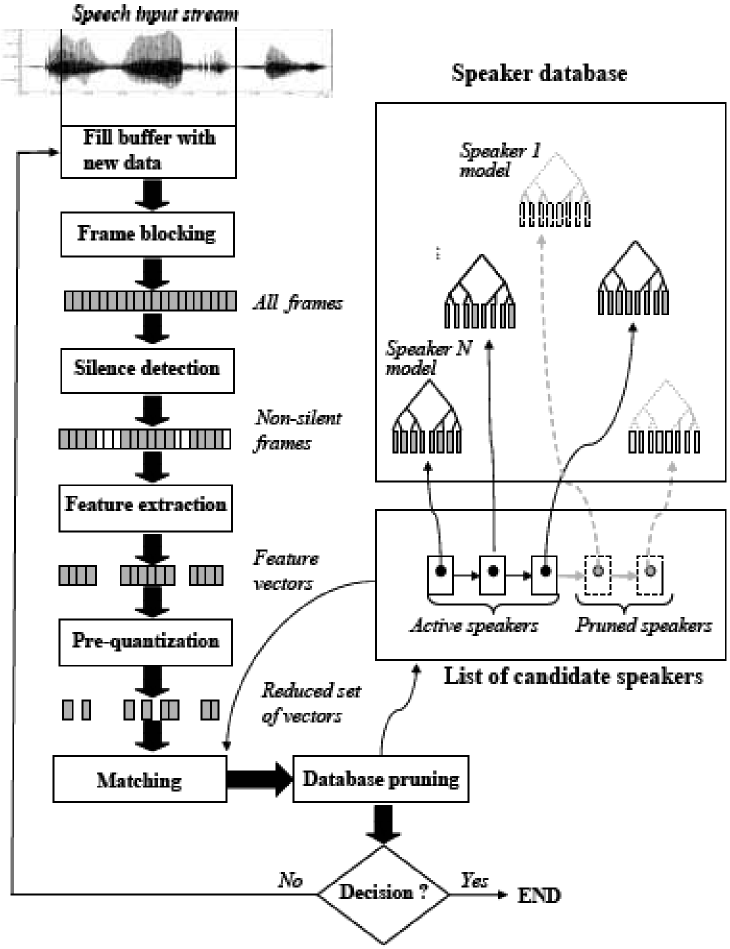


Figure 4.2 Diagram of the real time identification system. [4][10]

4.3.3 Speaker pruning

The idea of speaker pruning is illustrated in Figure 4.3. We must decide how many new (non-silent) vectors are read into the buffer before next pruning step. We call this the *pruning interval*. We also need to define the *pruning criterion* Figure 4.3 shows an example how the quantization distortion develops with time. The bold line represents the correct speaker. In the beginning, the match scores oscillate, and when more vectors are processed, the distortions tend to stabilize around the expected values of the individual distances because of the averaging in. Another important observation is that a small amount of feature vectors is enough to rule out most of the speakers from the set of candidates.

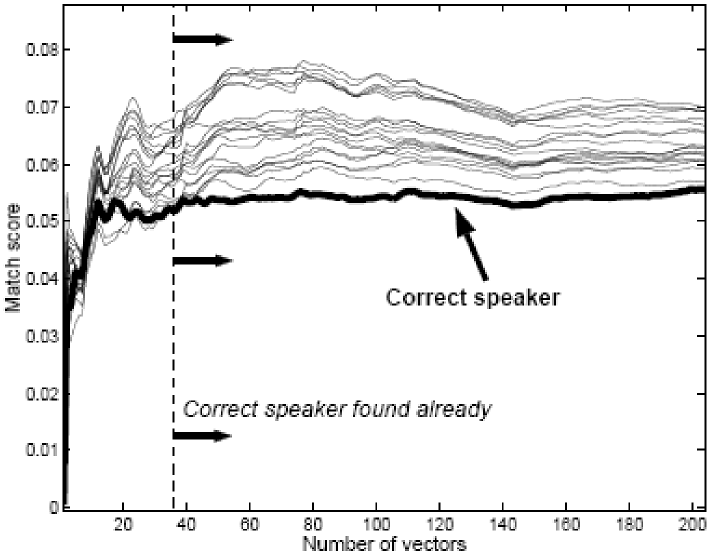


Figure 4.3 Illustration of match score saturation. [4][10]

4.4 Principles of Speaker Identification

Speaker identification further divides into two subcategories, which are text- dependent and text- independent speaker identification. Text- dependent speaker identification differs from text- independent because in the aforementioned the identification is performed on voiced instance of a specific word, whereas in the latter the speaker can say anything.

At the highest level, all speaker identification systems contain two main modules: Feature extraction and feature matching. Feature extraction is the process that extracts small amount of data from the voice signal that can later be used to represent each speaker. Feature matching involves the actual procedure to identity the unknown speaker by comparing extracted features from his/her voice input with the ones from a set of known speakers. Automatic speaker identification work is based on the premise that a person's speech exhibits characteristics that are unique to the speaker (Figure 4.4). However, this task has been challenged by the highly variant nature of input speech signal. The principles source of variance is the speaker itself, speech signals in training and testing sessions can be greatly different due to many facts such as people voice change with time, health condition (e.g. the speaker has a cold), speaking rates, etc. There are also other factors, beyond speaker variability, that present a challenge to speaker recognition technology, example of these are acoustical noise and variants in recording environments (e.g. speaker uses different telephone handsets).

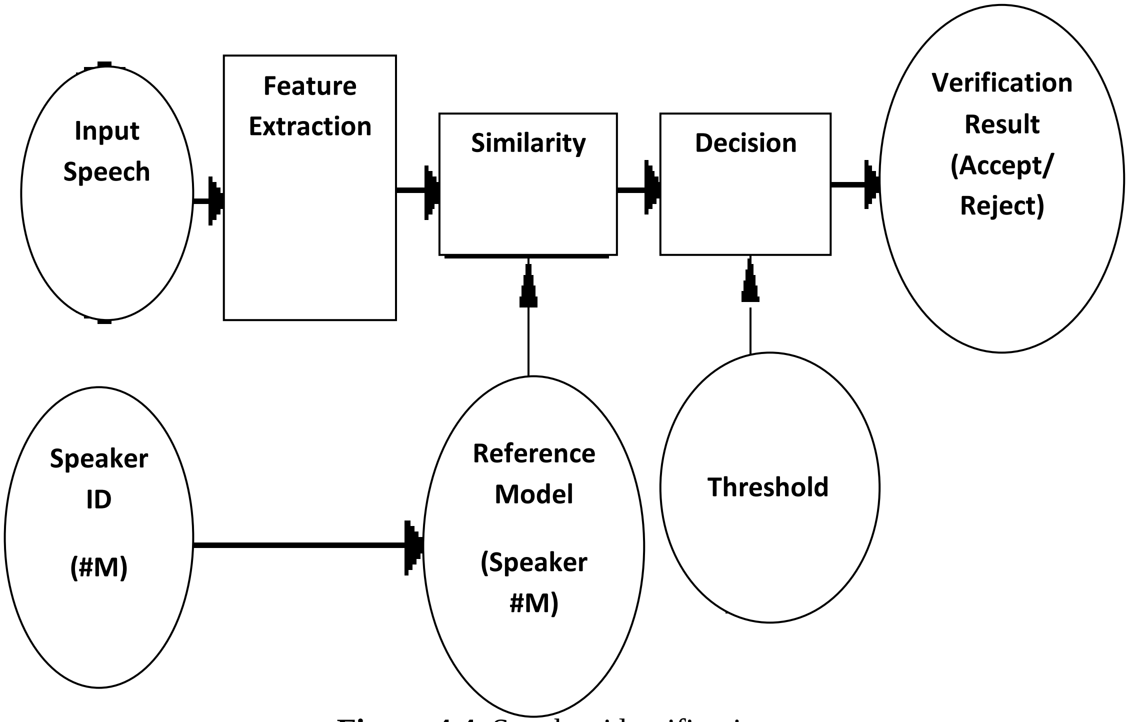


Figure 4.4 Speaker identification.

4.5 Verification versus Identification

Speech recognition, verification or identification systems work by matching pattern generated by the signal processing front-end with pattern previously stored or learnt by the speakers.

Voice based security systems come in two flavours, Speaker Recognition and Speaker Verification. In speaker recognition voice samples are obtained and features are extracted from them and stored in database. These samples are compared with various other stored ones and using methods of pattern recognition the most probable speaker is identified. As the number of speakers and features increases this method becomes more taxing on the computer, as the voice sample needs to be compared with all other samples stored. Another drawback is that when number of users increases it becomes difficult to find unique features for each user, failure to do so may lead to wrong identification.

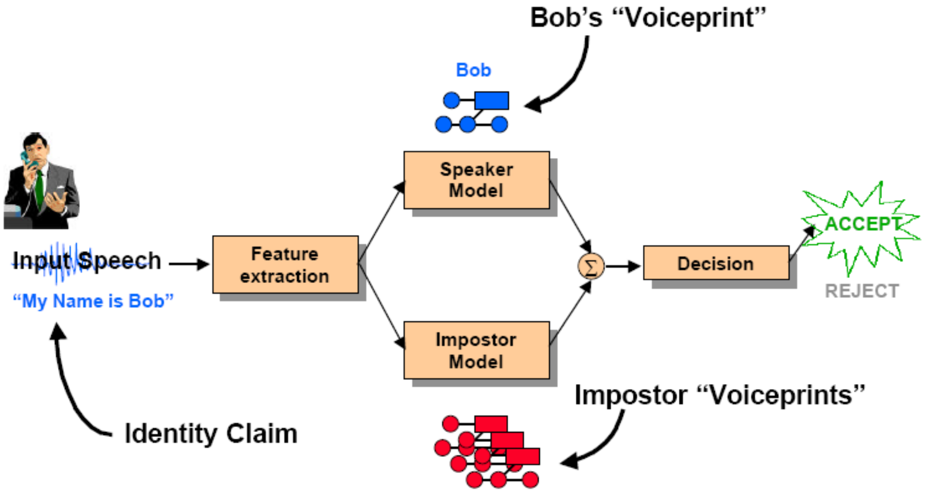


Figure 4.5 Components of speaker verification system. [20]

Speaker Verification is a relatively easy procedure wherein a user supplies the speaker's identity and record his voice. The goal of speaker verification is to confirm the claimed identity of a subject by exploiting individual differences in their speech. The features extracted from the voice sample are matched against stored samples corresponding to the given user, therefore verifying the authenticity of the user. In most cases a password protection accompanies the speaker verification process for added security.

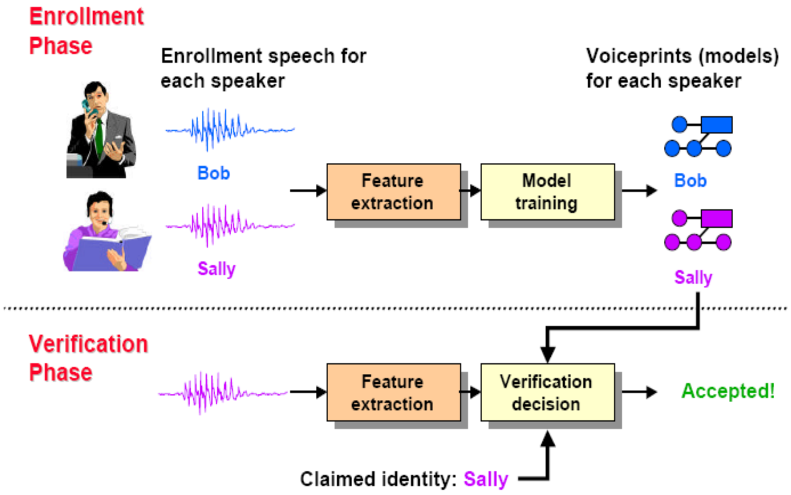
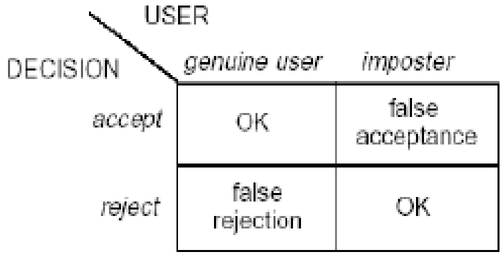


Figure 4.6 Two distinct phases to any speaker verification system. [20]

It is possible to expand the number of alternative decision from accept and reject into accept, reject and “unsure”. In this case the system has a possibility to be “unsure”, the user could be given a second chance.



DECISION \ USER	*genuine user*	*imposter*
accept	OK	false acceptance
reject	false rejection	OK

Figure 4.7 The decision matrix for the system. [22]

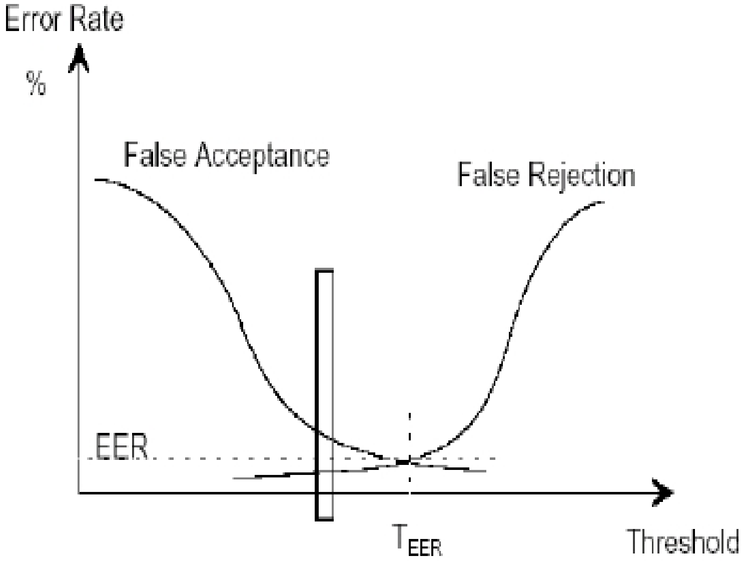


Figure 4.8 Threshold selection for minimizing errors in speaker verification. The system needs to work in small window, thus rendering the process as a sensitive one. [22]

4.6 Steps in Speaker Recognition

The most important parts of a speaker recognition system are the *feature extraction* and the *classification* method. The aim of the feature extraction step is to strip unnecessary information from the sensor data and convert the properties of the signal which are important for the pattern recognition task to a format that simplifies the distinction of the classes. Usually, the feature extraction process reduces the dimension of the data in order to avoid the "curse of dimensionality". The goal of the classification step is to estimate the general extension of the classes within feature space from a training set.

4.6.1 Extraction feature

Often, mel frequency cepstral coefficients (MFCC) are used. These features are well-known in the field of *speech* recognition also, therefore, they can be regarded as the "standard" features in speaker as well as speech recognition. However, experiments show that the parameterization of the MFC coefficients which is best for discriminating speakers is different from the one usually used for speech recognition applications. For example, speaker recognition error rates might be reduced if the "standard" MFCC feature dimension for speech recognition is increased.

The feature recognition process cuts the digitized audio signal, i.e. the sequence of sample values, into overlapping windows of equal length. The cut-out portions of the signal are called "frames", they are extracted out of the original signal every 10 or 20 ms. The length of a frame is about 30ms. For speaker recognition tasks, sometimes longer frames are used in comparison to the feature extraction method used for speech recognition in order to increase spectral resolution. Each frame in the time domain is transformed to a MFCC vector. Therefore, the original speech signal is converted into a sequence of feature vectors, each vector representing cepstral properties of the signal within the corresponding window. The feature vector sequences of training and test utterances are the inputs of the classification step of a speaker recognition system, which is now described in more detail.

4.6.2 Classification

In regard to the choice of the classification method, the kind of application of the speaker recognition system is crucial. For text independent recognition, speaker specific vector quantization codebooks or the more advanced Gaussian mixture models are used most often. For text dependent recognition, dynamic time warping or hidden markov models are appropriate.

4.6.2.1 Text independent recognition

Vector quantization is a technical which is also used for speech coding. The training material is used to estimate a codebook. This includes mean vectors of feature vector clusters which are given indices in order to identify them. For compression of speech, the index number of nearest cluster is used instead of the original feature vector. In order to be able to reconstruct the original signal, a revertable feature computation method has to be chosen (i.e. the MFCC features describes above cannot be used for speech coding). The quantization error in feature space is the mean distance between original feature vectors and nearest feature vector (i.e. the feature used for reconstruction). Obviously, the quantization error depends on the similarity between training material used for estimation of the codebook and the audio signal that is compressed. For example, if a code book is trained using speech signals, the compression of music with this codebook will result in a poor reconstruction for listener as well as in regard to the quantization error.

This observation is also true in regard to speaker specific codebooks which are used for speaker recognition. The training material of a speaker is used to estimate a codebook, which is the model for that speaker. The classification of unknown test signals is based on the quantization error. For example, for identification decision, the error of the test feature vector sequence in regard to all codebooks are computed. The “winner” is the speaker which code book has the smallest error between the test vectors and the corresponding nearest codebook vector.

Gaussian mixture models (GMM) are similar to codebooks in the regard that clusters in feature space are estimated as well. In addition to the mean vectors, the covariances of the clusters are computed, resulting in a more detailed speaker model if there is a sufficient amount of training speech.

4.6.2.2 Text dependant recognition

Dynamic time warping (DTW) stores the labeled training vector sequence without any further processing [3]. A test vector sequence is aligned to each of the training sequences such that a certain distance measure is minimized. Therefore, the classification algorithm can handle variations in regard to the length of the phonemes an utterance consist of.

Finally, a hidden markov model (HMM) [12] is a statistical model which may be used for text dependent recognition of speakers. Roughly speaking, they can be viewed as a combination of the DTW and the GMM approach. A HMM has a number of states which model distinct parts of, for example, a user's password for a pass- phrase authentication system. The feature vectors which are observed for the appropriate part of the pass phrase in training are used to estimate a density function. This is called "output density" of the HMM state. A hidden markov model is a more advanced representation for the pass phrase of a certain speaker, as the characteristic features for the phonemes that are present in the utterance are modeled statistically. Nevertheless, the DTW approach may be a better choice for a real-world speaker recognition system if the amount of available training data is not sufficient in order to reliability estimate the HMM's output densities.

Various typed of systems in use are:

1) *Fixed password system:* where all users share the same password sentence. This kind of system is a good way to test speaker discriminability in a text- dependent system.

2) *User- specific text- dependent system:* where every user has his own password.

3) *Vocabulary- dependent system:* where a password sequence is composed from a fixed vocabulary to make up new password sequence.

4) *Machine- driven text- independent system:* where the system prompts for an unique text to be spoken.

5) *User- driven text- independent system:* where the user can say text he wants

The first three are examples of text dependent systems, while the last two are text independent systems. We employed the first system, due the ease of implementation.

While the expression pairs open / closed set and identification / verification respectively can be used for other biometric authentication methods like, for example, iris, face or fingerprint as well, text dependent or independent recognition is obviously a specific characterization of a speaker recognition system.

The task of automatic speaker recognition is a classical example of a pattern recognition problem, which in general finds some kind of patterns within some real world sensor data. For all problems of pattern recognition, a training phase is required. For the example of a speaker authentication system, valid users of the system need to be enrolled. During the enrollment procedure, the system "learns" the person it is supposed to recognize. Speech samples of the user are required for this training phase.

During the later recognition process, the system compares another recorded speech signal (called test data) to the training utterance(s). The desired output of the system is the name of one of the training speakers, or a rejection if the test utterance stems from an unknown person.

Application dictates different speech modalities:

• Text-dependent recognition
– Recognition system knows text spoken by person
– Examples: fixed phrase, prompted phrase
– Used for applications with strong control over user input
– Knowledge of spoken text can improve system performance

• Text-independent recognition
– Recognition system does not know text spoken by person
– Examples: User selected phrase, conversational speech
– Used for applications with less control over user input
– More flexible system but also more difficult problem
– Speech recognition can provide knowledge of spoken text

4.7 Summary

This Chapter has described the classification of speaker recognition such a speaker identification and speaker verification and the methods of the speaker recognition such as text- independent and text- dependent and various speaker identification systems and the steps involved in a typical speaker identification process.