Faculté Polytechnique de Mons

Service de Théorie des Circuits et de Traitement du Signal

Confidence Measures for Speech/Speaker Recognition and Applications on Turkish LVCSR

Erhan MENGUSOGLU

Thèse originale acceptée par la Faculté Polytechnique de Mons pour l'obtention du grade de Docteur en Sciences Appliquées, le 20 avril 2004

Membres du jury:

Professeur J. TRECAT - FPMs - Président
Professeur J. HANTON - FPMs - Doyen
Professeur H. LEICH - FPMs - Promoteur
Docteur B. GOSSELIN - FPMs
Docteur J-C. FROIDURE - Multitel ASBL
Professeur B. MACQ - U.C.L.
Professeur F. GRENEZ - U.L.B.

Abstract

Confidence Measures for Speech/Speaker Recognition and Applications on Turkish LVCSR

by

Erhan MENGUSOGLU

Confidence measures for the results of speech/speaker recognition make the systems more useful in the real time applications. Confidence measures provide a test statistic for accepting or rejecting the recognition hypothesis of the speech/speaker recognition system.

Speech/speaker recognition systems are usually based on statistical modeling techniques. In this thesis we defined confidence measures for statistical modeling techniques used in speech/speaker recognition systems.

For speech recognition we tested available confidence measures and the newly defined acoustic prior information based confidence measure in two different conditions which cause errors: the out-of-vocabulary words and presence of additive noise. We showed that the newly defined confidence measure performs better in both tests.

Review of speech recognition and speaker recognition techniques and some related statistical methods is given through the thesis.

We defined also a new interpretation technique for confidence measures which is based on Fisher transformation of likelihood ratios obtained in speaker verification. Transformation provided us with a linearly interpretable confidence level which can be used directly in real time applications like for dialog management.

We have also tested the confidence measures for speaker verification systems and evaluated the efficiency of the confidence measures for adaptation of speaker models. We showed that use of confidence measures to select adaptation data improves the accuracy of the speaker model adaptation process.

Another contribution of this thesis is the preparation of a phonetically rich continuous speech database for Turkish Language. The database is used for developing an HMM/MLP hybrid speech recognition for Turkish Language. Experiments on the test sets of the database showed that the speech recognition system has a good accuracy for long speech sequences while performance is lower for short words, as it is the case for current speech recognition systems for other languages.

A new language modeling technique for the Turkish language is introduced in this thesis, which can be used for other agglutinative languages. Performance evaluations on newly de-

fined language modeling techniques showed that it outperforms the classical n-gram language modeling technique.

To my parents

To my wife, Nuran, for her tolerance

Acknowledgements

First of all, I would like to thank my thesis advisor Prof. H. Leich for his support and for providing me the research environment I needed for the realization of this thesis. I would also like to thank Prof. J. Hancq, the head of Signal Processing and Circuit Theory (TCTS) department at Faculté Polytechnique de Mons for the financial support provided through the the project ARTHUR financed by Wallonia Region of Belgium and D. Derestiat, the director of Multitel Research Center for the support provided during the last months of the work.

I acknowledge the guidance and suggestion of ASR group of TCTS and Multitel.

I would like to thank Christophe Ris and Laurent Couvreur for useful discussions and for sharing their experiences with me.

I would like to thank Olivier Deroo of Babel Technologies for his guidance in the earlier stages of the thesis work.

I would also like to thank Prof. Thierry Dutoit for his useful remarks on the thesis.

Finally, this thesis is dedicated to my wife, Nuran, and to my parents, Cemal and Cevahir.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> Men think they can copy Nature as Correctly
> as I copy Imagination; this they will find
> Impossible, & all the Copies or Pretended
> Copies of Nature, from Rembrandt to Reynolds,
> Prove that Nature becomes to its Victim nothing
> but Blots and Blurs...
> Copiers of Nature [are] Incorrect,
> while Copiers of Imagination are Correct.
>
> William Blake, *c. 1810*, "Public Address"

Speech recognition is shortly defined as finding the word content of the speech. There have been considerable advances in last decade in this research area. The main research directions in speech recognition are, robustness to noise and speaker variabilities, speaker identification/verification and dialogue which include interactive human-computer communication. Confidence measures are important for all fields of speech recognition.

Confidence is defined as a feeling or consciousness of reliance on one's circumstances. The confidence in speech recognition system is related to the reliability of the system. If we give a measure of confidence for the results of a speech recognition system, they will be more useful and better interpretable for the user.

The earlier uses of confidence measures in the literature were for detection of Out-Of-Vocabulary words detection and utterance verification in speech recognition [1], [2]. Utterance verification, noise cancellation, dialogue management, speaker verification, unsupervised speaker adaptation are principle domains for use of confidence measures.

Statistical analysis is the most important part of speech recognition. The statistical nature of speech recognition will be explained in the next chapter. Because of this property of speech recognition, hypothesis tests and confidence measures are essential to make accuracy assumptions on speech recognition results. From this point of view, confidence measures can be defined as "posterior probability of word correctness". Theoretical background of confidence measures and different uses of them will be discussed in detail in chapters 5, 6 and 7.

1.1 Thesis Perspectives

In this thesis we introduced new approaches in three fields, Turkish Large Vocabulary Speech Recognition (LVCSR), confidence measures for speech recognition and confidence measures for speaker recognition. The experiments are realized to evaluate the efficiency of new approaches.

The Turkish language has different characteristics than European languages, which require different language modeling techniques [48]. Since Turkish is an agglutinative language, the degree of inflection is very high. The new approach introduced in this thesis use the inflectional property of the Turkish language to improve efficiency of language modeling.

Turkish is one of the least studied language in the speech recognition field. Very first step of our work for Turkish LVCSR was to prepare a database. A database containing isolated and continuous speech data has been collected from the speakers of different age groups. The database is designed to be phonetically rich and to cover different topics collected from newspaper articles. After the data collection phase, acoustic modeling for Turkish speech recognition is obtained by using the Speech Training and Recognition Unified Tool (STRUT) [1].

Confidence measures can be defined for either acoustic model or language model. In this thesis we are focused on improvements in acoustic modeling based confidence measures. Statistical modeling techniques are used to train acoustic models. We introduce acoustic prior information usage for confidence measures and evaluate the efficiency of this new technique.

We define likelihood ratio based confidence measures for speaker verification and define a new metric for the interpretation of confidence measures. This new metric is based on the Fisher's z-transformation used generally to find a confidence interval for correlation coefficient. Confidence measures for speaker verification are used for speaker model adaptation purposes and the efficiency is tested in different testing environments.

1.2 Thesis Outline

This thesis starts with the definition of the theory of the speech/speaker recognition followed by Turkish LVCSR and definitions of some statistical methods related to the techniques used in the thesis. The results of the confidence measures for speech/speaker recognition tasks are followed by a conclusion.

Chapter 2 contains the definitions of speech recognition techniques. Speech recognition starts with acquisition of speech data followed by feature extraction. Basic feature extraction techniques are introduced in this chapter. Acoustic modeling techniques, dynamic time warping, vector quantization, hidden Markov modeling techniques, neural networks and hybrid

[1] http//tcts.fpms.ac.be/asr/strut.html

neural network hidden Markov models are explained. This chapter ends with definitions of basic language modeling techniques and a classification of speech recognition systems.

Chapter 3 is about speaker recognition techniques. It starts with an introductory level definition of other biometric technologies and locates speaker recognition in biometric recognition methods. Different speaker modeling techniques are defined with the advantages and disadvantages of using these techniques in different applications. Chapter ends with definitions about use of confidence measures for model adaptation.

Chapter 4 starts with characteristics of the Turkish language that are important for speech recognition. The morphology of the Turkish language and its difference compared to European languages is given later in the chapter. The steps of Turkish speech database preparations and the characteristics of the database are given. The new language modeling technique introduced by this thesis is explained in the end of this chapter.

Chapter 5 contains definitions of different types of confidence measures for speech recognition and the new confidence measure defined in this thesis. Use of confidence measures in different speech recognition tasks is given. Some experiments for confidence measures on Phonebook database and newly created Turkish database are given in the end of the chapter. This chapter ends with results and some conclusions.

Chapter 6 is about confidence measures for speaker recognition tasks. Speaker verification and then application of confidence measures are explained in detail. Transformed confidence measures which provide a better interpretation of confidence measures are studied.

Chapter 7 gives speaker adaptation methods for speaker verification. Use of adaptation techniques and efficiency of confidence measures are tested in this chapter.

The last chapter, Chapter 8, contains a conclusion about the techniques used in this thesis and the results obtained for different techniques.

Chapter 2

Speech Recognition

Speech recognition is the identification of a portion of speech by a machine. The speech is digitized and compared with some coded dictionaries to identify the word context of it.

The system which is used for this purpose is called speech recognition system. This system must be first "trained" using a speech corpus which represents the words to be recognized. This training speech can include all the dictionary words in case of small dictionary but when large dictionaries are to be recognized the content of training speech must be carefully determined since it is difficult to have all the vocabulary words in the training corpus. In this case the training corpus must be large enough to include all sub-word units (syllables, triphones, etc.) included in the recognition dictionary.

In this chapter, in order to give a better description of speech recognition, the process is broken down into subsections. First, human speech production system will be explained. Then, speech communication and hearing will be discussed to complete the source-channel model of speech recognition. After the definition of source-channel model, speech recognition process will be decomposed into its parts and methods used for each part will be addressed.

2.1 What is Speech?

The basic definition for speech must be based on voice. Voice (or vocalization) is defined as the sound produced by humans and other vertebrates using the lungs and the vocal folds in the larynx, or voice box. Voice is not always produced as speech, however. Infants babble and coo; animals bark, moo, whinny, growl, and meow; and adult humans laugh, sing, and cry. Voice is generated by airflow from the lungs as the vocal folds are brought close together. When air is pushed past the vocal folds with sufficient pressure, the vocal folds vibrate. If the vocal folds in the larynx did not vibrate, speech could only be produced as a whisper. The voice is as unique as fingerprint. It helps define personality, mood, and health.

Speech is the voice carrying an idea with the help of a language. Many things have to happen for us to speak:

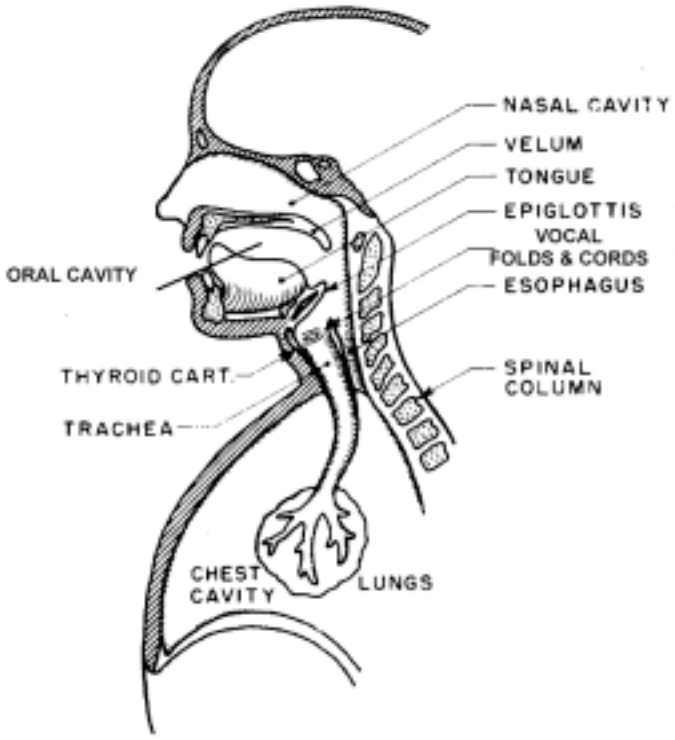

Figure 2.1: Human voice production system

- The person must have a thought or idea, want or need that must be communicated to another person.
- The idea must be sent to the mouth with instructions on which words to say and which sounds make up those words.
- The brain must send signals to the muscles that produce speech - the tongue, lips and jaw.
- The lungs must have enough air to force the vocal chords to vibrate.
- The body parts must be co-ordinated and strong enough to create the right sounds and words.
- The sounds must be clear enough to form words that other people can understand.
- There must be another person to receive the communication and respond.

The main parts of human voice production system are shown on Figure 2.1. Brain (as the source of ideas), lips and teeth must be added to these parts to product speech. By a more general definition, speech is simply the acoustic wave that is radiated from the vocal system when air is expelled from the lung and the resulting flow of air is perturbed by a constriction

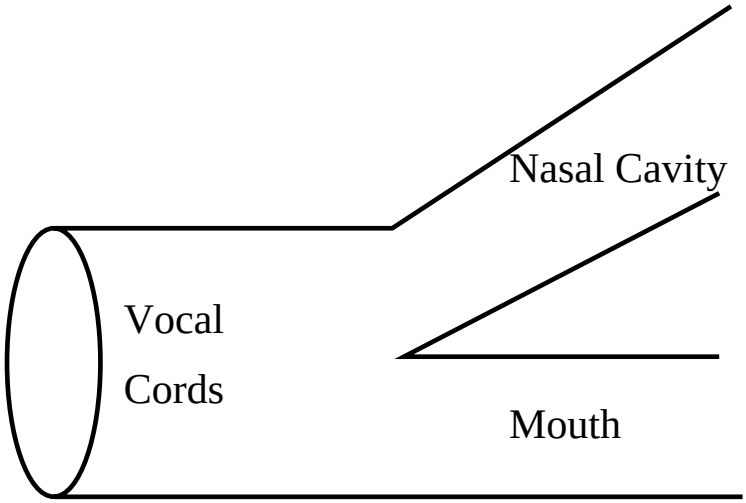

Figure 2.2: Source-channel model for vocal tract

somewhere in the vocal tract. Vocal tract includes all parts shown on Figure 2.1 including also lips and teeth.

Humans express thoughts, feelings, and ideas orally to one another through a series of complex movements that alter and mold the basic tone created by voice into specific, decodable sounds. Speech is produced by precisely coordinated muscle actions in the head, neck, chest, and abdomen.

Speech production starts with lungs. When the air is pushed up and out of the lungs, it passes through the trachea and vocal folds into the larynx. During breathing, vocal cords are open and airflow passes between left and right vocal cords. To product speech, the gap between vocal cords becomes narrower and this results in vibration of vocal cords which produce a sound. The airflow is then interrupted periodically by opening and closing of this gap. This process modify the frequency of the sound produced. When the vocal cords are strained and the pressure of the air from lungs is high, the vibration period becomes short and this results in high frequency sounds. Conversely, when the gap between vocal cords is greater and the air pressure is low, the resulting sound has a low frequency.

Once the sound is produced by the vocal cords, it passes through vocal tract. Vocal tract is the main part of human speech production system. As described above, speech is the result of perturbations applied to the sound generated by vocal cords when it passes through vocal tract. A minimalist model of vocal tract is shown on Figure 2.2. In this figure, vocal tract is shown as a combination of two tubes. One of them is mouth and the other one is the nasal cavity which ends by nostrils. The sound is modified when it passes from these tubes and diffused by air when it leaves the mouth and nostrils.

The vocal cord vibration frequency is the fundamental frequency. The sound source, containing the fundamental and harmonic components, is modified by vocal tract to produce

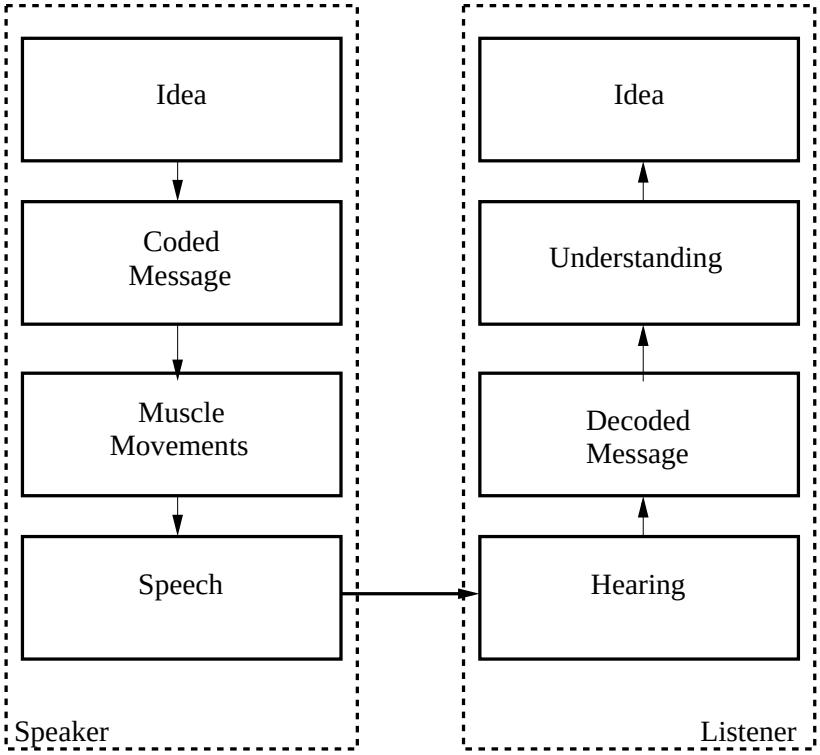

Figure 2.3: Speech communication between humans

different phonemes. During vowel production, the vocal tract remain in a relatively stable configuration. For the production of fricatives and plosives, air flow is altered by the tongue or lips. For each phoneme, different parts of vocal tract are used to alter the sound produced from vocal cords. The result is a sequence of phonemes which forms words and then phrases. The process of generation of speech from vibration of vocal cords is called "articulation".

2.2 Speech Communication between Humans

The main purpose of speaking is to communicate. Humans communicate with other humans in many ways, including body gestures, printed text, pictures, drawings, and speech. But surely speaking is the most widely used in our daily affairs. The basic model of speech communication between humans is shown on Figure 2.3. This figure shows one way communication between two person. There is one speaker and one listener. The aim is to convey the speakers idea to the listener. First the idea is formulated as a message using a special coding mechanism which is called "language". Then the coded message is transformed into speech using some muscle movements which generate some sounds through articulation process. The speech is acquired by the listener through hearing process. Then the message is decoded into language codes; after combination of these codes through understanding process, the main idea which the speaker wanted to pass is understood.

For better understanding of sound production in a language, the language can be divided into different coding levels. The main coding levels are sentence, word, morpheme and phoneme. Sentence is the smallest unit that carry an idea. Word is the smallest unit that has a meaning and that can occur by itself. Morpheme is the smallest distinctive unit that cannot occur by itself unless it is in a monomorphemic word. Phoneme is the smallest contrastive unit in the sound system of a language. Phonetic realization of a morpheme is a morph, phonetic realization of a phoneme is a phone. It is possible to have two or more different phonetic realization for morphemes and phonemes, those are allomorphs and allophones.

The most basic way to organize speech sounds (phones) is to separate them into two main group according to the level of constriction on the vocal tract. When there is little or no constriction the realized phones are vowels, and when there is a total or moderate constriction the phones are consonants.

Consonants can be differentiated by reference to three parameters; place of articulation, manner of articulation, and voicing.

Place of Articulation, is the point in the vocal tract where the constriction is applied.

1. Bilabial, two lips together are used to realize the phoneme.

2. Labiodental, the upper teeth contact or approach the lower lip.

3. Dental, the tongue is touching or approaching the back of the teeth.

4. Alveolar, the front of the tongue touches or approaches the alveolar ridge.

5. Palatal, the tongue is touching the roof of the mouth but a bit further than the "alveolar". The touching point is hard palate.

6. Velar, the back of tongue rises high enough to touch the velum.

7. Glottal, the vocal folds are drawn close enough together to produce a hissing or whispering sound.

Manner of Articulation, is the type of constriction applied at a place of articulation.

1. Stop, the vocal tract is momentarily closed.

2. Fricative, the constriction is not total, there is a small channel which allow air to rush

3. Affricate, the stop and the fricative manners are combined into a single new type.

4. Nasal, the flow of air is blocked in the mouth but allowed to flow freely through the nasal cavity

5. Liquid, the air flow around the obstruction is quite free.

6. Glide, the most vowel-like consonants. There is almost no obstruction on air flow as in vowels.

Voicing, is the third dimension of articulation. There are some consonants that have the same place and manner of articulation but different voicing properties. In voiced consonants there is a vibration in vocal cords while there is no vibration in voiceless consonants. Classification of English consonants by examples is shown in Table 2.1.

Table 2.1: Classification of English consonants according to the constrictions on the vocal tract.

Manner of articulation	Voicing	Place of Articulation						
		Bilabial	Labiodental	Dental	Alveolar	Palatal	Velar	Glottal
Stops	Voiceless	**p**at			**t**ack		**c**at	
	Voiced	**b**at			**d**ig		**g**et	
Fricatives	Voiceless		**f**at	**th**in	**s**at	fi**sh**		**h**at
	Voiced		**v**at	**th**en	**z**ap	a**z**ure		
Affricates	Voiceless					**ch**ur**ch**		
	Voiced					**j**udge		
Nasal		**m**at			**n**at		si**ng**	
Liquids					**l**ate	**r**ate		
Glides		**w**in				**y**et		

Table 2.2: Classification of English vowels according to the position of the tongue.

	Front	Center	Back
High	b**ee**t		b**oo**t
	b**i**t		b**oo**k
	b**a**by		b**o**de
		sof**a**	
Middle	b**e**t		b**ou**ght
	b**a**t	b**u**t	
Low			
			p**a**lm

Vowels are voiced and there are little or no constriction of the vocal tract during vowel production. The differences between vowels are mainly obtained by changing the position of the tongue in the mount. Table 2.2 gives the classification of vowels according to the position of the tongue.

2.2.1 Hearing

According to the Figure 2.3, speech communication include two parties; speaker and listener. The first part of this section explain speech production. Hearing is the other important function helping speech communication between humans. The first part of hearing process is the "ear". Human ear is fully developed at birth and responds to sounds that are very faint as well as sounds that are very loud. The ability to hear is critical to the attachment of meaning to the world around us.

Hearing mechanism is consisted in five sections:

- Outer ear

- Middle ear
- Inner ear
- Acoustic nerve
- Brains' auditory processing centers

The **outer ear** consists of the *pinna* and the ear canal. *Pinna* is the part we can see from outside. It serves as a collector of sound vibrations and funnels the vibrations into the ear canal. It helps us to localize the sound. Ear canal carries the sound vibration to middle ear by filtering first the foreign bodies from air.

The **middle ear** begin with the eardrum at the end of ear canal. It contains three tiny bones called the *ossicles*. These bones form a connection from eardrum to the inner ear. They convert the vibrations on eardrum to mechanical vibrations.

The **inner ear** contains the sensory organs for hearing and balance. The *cochlea* is the hearing part of the inner ear. It is a bony structure shaped like a snail and filled with liquid. The *Organ of Corti* is the sensory receptor inside the cochlea which holds nerve receptors for hearing called the *hair cells*. The mechanical energy from movement of the middle ear moves the cochlea's fluid which then stimulate tiny hair cells. Individual hair cells respond to specific sound frequencies. That means each frequency component of the sound stimulate certain hair cells. They transmits the stimulations then to the brain by acoustic nerve.

The **acoustic nerve** carries impulses from cochlea to the brain. Nerve fibers from each ear divide into two pathways which carry impulses to two sides of brain. Each side of brain receive impulses from both ears.

The **central auditory system** deals with the processing of auditory information as it is carried up to the brain. The tasks are:

- Sound localization
- Auditory discrimination
- Recognizing patterns of sounds
- Time aspects of hearing like temporal ordering etc.
- Deciding the quality of the sound and reducing the auditory performance in presence of noise

Once the hearing process is completed, brain try to decode and understand the message. Decoding and understanding is done according to previous training of the listener. Listener can understand the message and can receive the idea transmitted by the speaker only if he/she is trained to understand the language spoken by the speaker.

2.3 Definition of Speech Recognition

Speech recognition is the process that allows humans communicate with computers by speech. This process can be simply shown by replacing the listener in Figure 2.3 by a computer. The purpose is to transmit the idea to the computer.

There are lots of other communication methods between humans and computers which require some input devices. Keyboards, mouses, touch screens are the most classical examples of input devices with high accuracies. Those input devices are not efficient enough in some conditions, especially when the use of hands is not possible. They need also a certain level of expertise for being used.

There are some other recent researches on human-computer interaction with brain waves but this research field is still in its beginning phase [3]. Since speech is the most natural way of communication between humans, it is important to make possible the use of speech to communicate with computers. By enabling speech recognition, communication between humans and computers can be easier, as easy as using a telephone or speaking to a microphone, and faster than the other alternatives like keyboards or touch screens.

There are many application areas for speech recognition. The main areas can be listed as, home use, office use, education, portable & wearable technologies, control of vehicles, avionics, telephone services, communications, hostile environments, forensics & crime prevention, entertainment, information retrieval, biometrics surveillance, etc.

Speech recognition is closely related to other speech related technologies, such as, automatic speech recognition, speech synthesis, speech coding, spoken language understanding, spoken dialogue processing, spoken language generation, auditory modeling, paralinguistic speech processing (speaker verification/recognition/identification, language recognition, gender recognition, topic spotting), speech verification, time-stamping/automatic subtitling, speech to speech translation, etc.

Speech recognition is a multi-disciplinary discipline spanning to acoustics, phonetics, linguistics, psychology, mathematics and statistics, computer science, electronic engineering, and human sciences.

Speech recognition has been a research field since 1950s. The advances are not satisfactory enough despite more than 50 year of research. This is mainly due to openness of speech communication to environmental effects and existence of various variabilities that are difficult to model in the speech. The speech is acquired by computers using microphones which record it as energy levels at certain frequencies. Since speech is passed through air before having recorded digitally, the recording contains environmental effects also. Speech recognition process is based only on the speech content of the recorded signal. The quality of the signal must be improved before speech recognition. Hermansky [4] claims that indiscriminate use of accidental knowledge about human hearing in speech recognition may not be what is needed. What is needed is to find the relevant knowledge and extract it before doing any

Figure 2.4: Speech recognition

further processing towards speech recognition.

Figure 2.4 shows the speech recognition process in a simplified way. Speech recognizer is a black box which contains the necessary information to recognize the speech at the input. At the input there should be a microphone and at the output there is a display to show the recognized speech.

The remaining part of this chapter defines the speech recognition cycle by decomposing it to its basic parts. In a more general context, speech recognition can be seen as a signal modeling and classification problem [5]. The aim is to create models of speech and use these models to classify it. The speech include two parts which can be modeled; **acoustic signal**, and **language**.

As a modeling problem, speech recognition includes two models.

- Acoustic Model
- Language Model

These two models will be explained later in detail. Acoustic model is the modeling of acoustic signal and it starts with acquisition of speech by computers. Language model is the modeling of speaker's language and it will be used at the end of classification process to restrict the speech recognition to extract only acceptable results from speech signal.

The "Speech Recognizer" black box on Figure 2.4 can be opened as on Figure 2.5 that shows, the main procedures in speech recognition are **Speech Acquisition**, **Feature Extraction** and **Classification**. Classification is sometimes called decoding. The most important parts which affects the performance of the system are **Acoustic Model** and **Language Model**. These models are obtained after a training procedure. Speech acquisition and feature extraction parts are also important for representing speech signal in classification phase.

2.4 Speech Acquisition

Speech acquisition includes converting the acoustic signal to some computer readable digital codes. This process can also be called as "digital recording".

Speech signal is an analog signal which has a level(loudness), shape, and frequency. The first thing to do with speech signal is to convert it from analog domain which is continuous to digital domain which is discrete. To convert a signal from continuous time to discrete time,

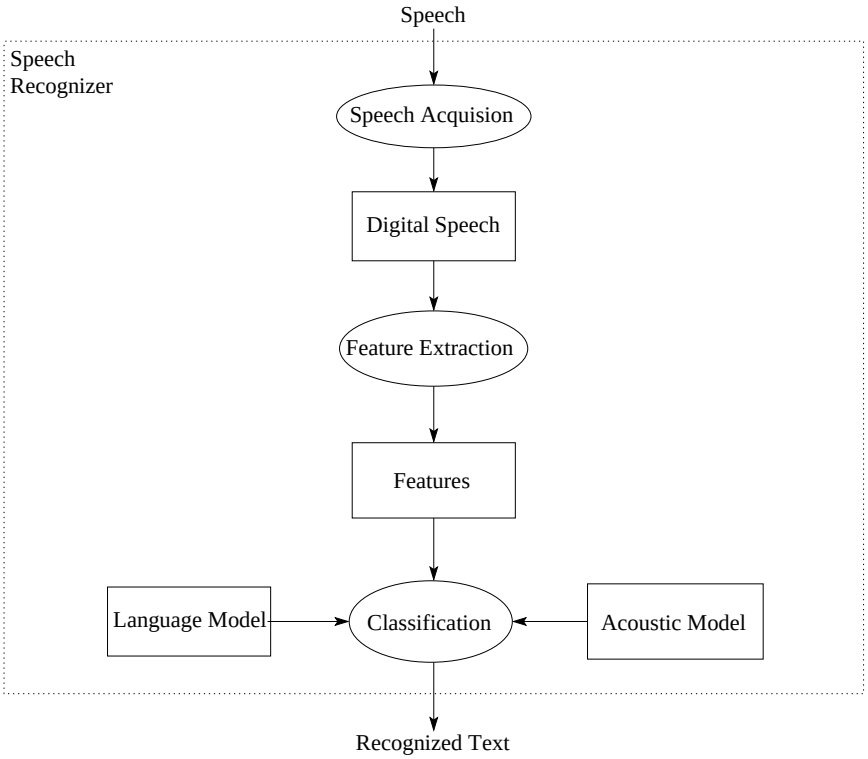

Figure 2.5: Speech recognition (detailed)

a process called sampling is used. The value of the signal is measured at certain intervals in time. Each measurement is referred to as a sample.

When the continuous analog signal is sampled at a frequency F, the resulting discrete signal has more frequency components than did the analog signal. To be precise, the frequency components of the analog signal are repeated at the sample rate. That is, in the discrete frequency response they are seen at their original position, and are also seen centered around +/- F, and around +/- 2F, etc.

If the signal contains high frequency components, we will need to sample at a higher rate to avoid losing information that is in the signal. In general, to preserve the full information in the signal, it is necessary to sample at twice the maximum frequency of the signal. This is known as the Nyquist rate.

Telephone speech is sampled at 8 kHz, that means the highest frequency represented is 4000 Hz which is greater than the maximum frequency standard for telephone in Europe (3400 Hz). A sampling frequency of 16kHz is regarded as sufficient for speech recognition. Generally, speech signal sampling frequency is chosen between 600 Hz and 16000 Hz. The frequency range that human ear can hear is between 80 Hz and 8000 Hz. The extreme limits are 20 Hz and 20 kHz. [6].

The level of sampled speech signal is the sampling resolution. Use of more bits gives better resolutions. For telephone speech compressed 8 bits sampling resolution is used. For

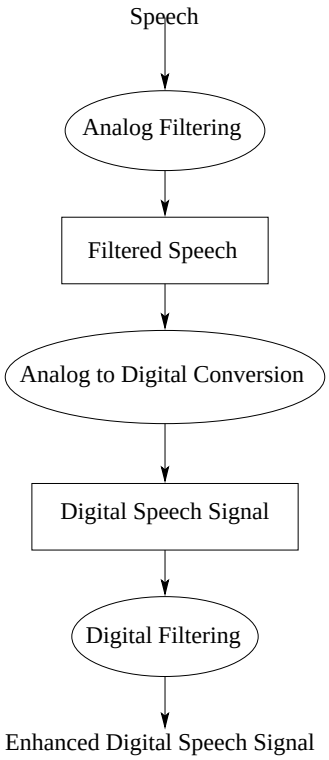

Figure 2.6: Acquisition of speech by computer for speech recognition

speech recognition, in general, a resolution of 12 bit is sufficient. For higher accuracies, we need to use more bits per sample.

The speech signal can contain some redundant frequency components which are considered as noise. Some of those frequencies can be filtered. Generally, filters are used to modify the magnitude of signals as a function of frequency. Desirable signals in one range of frequencies (usually called a band) are passed essentially unchanged, while unwanted signals (noise) in another band are reduced (attenuated).

Figure 2.6 shows the structure of a speech acquisition block which can be integrated into speech recognizer in which the analog filtering part is generally integrated into a microphone. A device is used to record the speech digitally according to sampling theory. Digitalized speech can than be filtered digitally to improve the quality of speech. Digital filtering process is generally integrated in "Feature Extraction" part of the speech recognizer.

The digital speech signal can have various formats. Digital representation of speech is generally called "coding". There are three groups of coding, namely **Waveform Coding**, **Source Coding** and **Hybrid Coding**.

The Waveform Coding attempts to produce a reconstructed signal whose waveform is as close as possible to the original. The resulting digital representation is independent of the type of signal. The most commonly used waveform coding is called "Pulse Code Modulation" (PCM). It is made up of quantizing and sampling the input waveform. There are two variants

of this coding method. Those are Differential PCM (DPCM) which quantizes the difference between two samples, and Adaptive DPCM (ADPCM) which tries to predict the signal and use a suitable quantization for different portion of that signal.

The Source Coding is model based. A model of the source signal is used to code the signal. This technique needs a priori knowledge about production of signal. The model parameters are estimated from the signal. Linear Predictive Coding (LPC) uses source coding method. The value of the signal at each sample time is predicted to be a linear function of the past values of the quantized signal.

The Hybrid Coding is a combination of two other coding methods. An example of this type of coding is "Analysis by Synthesis". The waveform is first, coded by source coding technique. Then the original waveform is reconstructed and the difference between original and coded signal is tried to be minimized.

2.5 Feature Extraction

Speech signal is considered to be produced by a non-stationary random process [6]. This characteristic of speech signal requires estimation of parameters over a relatively important time intervals, like tens of seconds, and for several speakers. Speech is also considered to be stationary on short-term intervals from 10 ms to 30 ms. Speech recognition technology is based on statistical processing of short-term characteristics of speech. Extraction of parameters about the characteristics of short-term speech portions is called "Feature Extraction". Feature extraction provide a parameterization of speech to be used in speech recognition. For effective performance of speech recognition, feature extraction should provide:

- Distinctive parameters for different speech units.
- Accurate reduction in dimension of parameters which will help easier statistical modeling.
- Redundancy in the features must be minimized. That means, only speech related informations must be contained in the features.
- Loss of speech related information must be minimized.

Feature extraction is called also as "the front-end", and is considered as independent from the other parts of speech recognition.

Feature extraction is generally based on short time spectral features of speech. Those features are based on either Fourier transformation or linear prediction. After spectral analysis usually there is a cepstral analysis. The most popular and performant feature extraction methods currently used for speech recognition are the *Mel-frequency cepstral coefficients(MFCC)* [7] and the *perceptual linear prediction* [8].

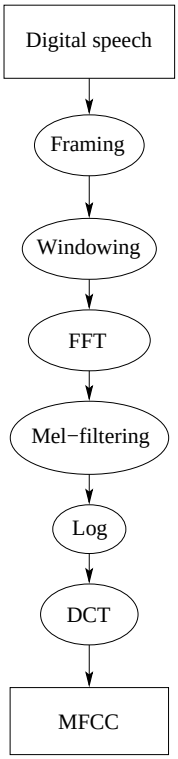

Figure 2.7: Block diagram of MFCC algorithm

MFCC is based on a perceptually scaled frequency axis. The mel-scale provides higher frequency resolution on the lower frequencies and lower frequency resolutions on higher frequencies. This scaling is based on hearing system of human ear. MFCC algorithm is shown an Figure 2.7. This algorithm gives a vector of n coefficients. ($n < 20$).

As shown in Figure 2.7 the first step of the MFCC algorithm is called "framing". At this step the time interval for the feature extraction is determined. This is done due to some predefined frame length. Generally a frame length of 10 ms to 30 ms is chosen for speech recognition. Overlapped framing is used for effective information extraction between two adjacent frames. That means, for example, a frame of 30 ms is shifted 10 ms to have a new frame, 20 ms of previous frame is included in new one. In "windowing" step a window function is applied to the frame. "Hamming" window is the most frequently used windowing technique for speech processing. It is defined by the following formula:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), 0 \leq n \leq N-1 \tag{2.1}$$

where N is the length in frame of the window and n is the frame index. The Fast Fourier Transformation (FFT) is then applied to the window to have the frequency content of speech signal in current frame. The frequencies are then filtered by mel-scale filter which is defined as:

$$Mel(f) = 2595\ log_{10}(1 + \frac{f}{700}) \tag{2.2}$$

where f is the frequency in Hz. The Discrete Cosine Transformation (DCT) is applied to the logarithm of the mel-scale filtered frequencies. The first N coefficients are selected as feature vector representing the selected frame.

PLP is based on linear prediction. The most performant variant of the PLP is the RASTA-PLP which include a filtering technique based on suppression of noise in the speech.

RASTA-PLP algorithm is defined as follows [8]:

1. Framing and windowing (Hamming window) the digital speech signal.
2. Compute power spectrum (FFT).
3. Map the power spectrum to an auditory frequency axis, by combining FFT bins into equally-spaced intervals (critical bands).
4. Take the logarithm.
5. Filter each frequency band with the RASTA filter as defined in [8].
6. Take the inverse logarithm.
7. Fix-up the auditory spectrum with equal-loudness weighting and cube-root compression.
8. Calculate cepstral coefficients by taking the DFT of the log of a set of spectral coefficients.

The extracted feature vectors for short term speech portions, are then used in next steps of speech recognition. The speech recognition can be now defined as a statistical pattern recognition problem. The patterns are feature vectors.

2.6 Statistical Pattern Recognition

A pattern is a set of observed information which can be measurements or features. In speech processing, a feature vector obtained after feature extraction process applied to speech signal is a pattern. Pattern recognition is defined also as the study of how machines observe the environment, learn to distinguish patterns of interest from background and make sound and reasonable decisions about categories of the patterns [9].

Pattern recognition is very important in communication of humans with machines. The real world observations are transfered to machines with the aid of sensors. The patterns can be from different sensor domains from radars to microphones or from cameras to electroensefalography devices. Automatic recognition of patterns obtained from sensors can help to solve various problems from a variety of scientific disciplines such as biology, psychology,

Table 2.3: Examples of pattern recognition[9]

Problem Domain	Application	Input Pattern	Pattern Classes
Bioinformatics	Sequence analysis	DNA/Protein sequence	Known types of genes/patterns
Data mining	Searching for meaningful patterns	Points in multidimensional space	Compact and well-separated clusters
Document classification	Internet search	Text document	Semantic categories
Document image analysis	Reading machine for the blind	Document image	Alphanumeric characters, words
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective/ non-defective nature of product
Multimedia database retrieval	Internet search	Video clip	Video genres
Biometric recognition	Personal identification	Face, iris, fingerprint, voice	Authorized users for access control
Remote sensing	Forecasting crop yield	Multi spectral image	Land use categories, growth pattern of crops
Speech recognition	Call center assistance, spoken dialog with computers	Speech waveform	Spoken words, text content of speech

medicine, marketing, computer vision, artificial intelligence, remote sensing etc. Table 2.3 gives some examples of contemporary pattern recognition fields.

The advances in sensor technologies and the increase in computing power have boosted the ability and diversity of pattern recognition tasks. Concurrently the demand for automatic pattern recognition is rising enormously with increasing database sizes. Since the world is getting more "digitized", the demand on automatized pattern recognition will increase more and more. This demand shows the importance of automatic pattern recognition. A typical pattern recognition system is composed of:

1. Data acquisition and preprocessing,
2. Data representation (feature extraction), and
3. Decision making (classification).

Each problem domain has its specific sensor(s) for data acquisition followed by a preprocessing. For speech recognition, microphones and acoustic coding like PCM are for this purpose. After this first step data representation or feature extraction is applied to obtain a reduced and easy to process set of parameters for the data. For example MFCC feature vectors for speech frames. The last step is to classify the data and make some decision about the content of collected data. Decision making step generally include a one time training phase in which some pre-classified data are used for training the decision making system.

Pattern recognition is based on two types of classification: (1) Supervised Classification; (2) Unsupervised Classification. In supervised classification, the data is classified into predefined classes and in unsupervised classification the aim is to cluster the data at the input. Speech recognition needs supervised classification since the collected data is classified into phonetic classes or word classes.

There are four best known approaches for pattern classification:

1. Template matching,
2. Statistical classification,
3. Syntactic or structural matching,
4. Neural networks

These methods are not necessarily independent from each other, it is possible to combine two or more of them to obtain a new and more performant classification method. This is the case, for example, for statistical classification and neural networks methods which are combined as a hybrid method and is shown to be the better performing speech recognition method. This method will be discussed in detail later in this chapter.

Template matching is the simplest and earliest approach to pattern classification. Matching is used to determine the similarity between two observations of the same type. In template matching, a template of pattern to be recognized is already available to the system. The pattern to be matched is compared with the stored template according to some distance (similarity) measure. This measure should be aware of scale changes, rotations or translations. Stored templates can be optimized with some training data before they are used for classification. When the number of classes increase and intra-class variability is high the performance of this method decreases. Dynamic Time Warping method which will be explained later in this chapter can be considered as a template matching method with some improvements on search capability.

Statistical approach is based on features. Each pattern is represented as a feature vector. Given a set of training feature vector, the purpose is to classify the feature vector into pre-defined class. Each class must be represented by sufficient amount of feature vectors in training data set. The class boundaries are determined statistically by probability distributions of the pattern belonging each class. The performance of this method depends on good representation of each class in training data with a sufficiently large database which cover all intra-class variations that may be present in each class. The performance of class boundary determination algorithm is also important for better classification results. Hidden Markov Model, which is explained later in this chapter, is a statistical classification method.

Syntactic approach is based on a hierarchical processing of subpatterns. Each pattern is composed of subpatterns and subpatterns are composed of simpler structures. There is a formal analogy between the syntax of the language which created the pattern and the structure of pattern. This method requires large amount of data to train the grammar for each pattern.

Neural networks attempt to use some organizational principles (learning, generalization, computation, etc.) in a network of weighted directed graphs. They are capable of learning complex nonlinear relationships between output and input through weight updating of graph nodes (neurons). Feed-forward network and multi-layer perceptron are commonly used neural networks for pattern classification tasks. The first step of pattern classification with neural networks is, as with the other pattern classification methods, training which is called learning in this context. In this step network weights are updated to have minimum classification errors according to some pre-classified training data.

Statistical pattern recognition can be modeled as a two-step process. The first step is training and the second is classification. In training step a classifier is trained according to some training algorithms. The feature space is partitioned and class boundaries are determined. In classification step each feature vector at the input is classified into a class according to class boundaries.

The classification task in pattern recognition can be defined as follows; let a given pattern

be represented by the feature vector x of d dimension

$$x = (x_1, x_2, x_3, ..., x_d)$$

and let the c classes be denoted by

$$W = (w_1, w_2, w_3, ..., w_c)$$

. The problem is to assign x to one of the classes in W. The features have a probability density function conditioned on the pattern class. A pattern x belonging to class w_i is an observation drawn randomly from the conditional probability function $p(x|w_i)$. The class boundaries for decision making are defined by decision rules. The most popular decision rules are Bayes decision rule, maximum likelihood rule and Neyman-Pearson rule. A complete review of pattern classification methods can be found in [9]. In speech recognition, generally, maximum likelihood and Bayes decision rule are frequently used. These methods will be explained later in the acoustic modeling section of this chapter.

The performance of classifier, as stated before, depends on training which proceeds the classification. Both the number of available number of samples and good representation of each class in the training data. Since the purpose of designing a classifier is to classify future samples which will be different than those exist in the training data, a classifier optimized for training samples may not work well for future samples. The classifier must have generalization ability which means it must work well for unseen test data. Overfitting of classifier to training data must be avoided.

2.7 Pattern Classification for Speech Recognition

Speech recognition include two pattern classification steps. The first one is acoustic processing which results a sequence of word or sub-word speech units. The output of first step is then used for language processing which guaranties a valid speech output within the rules of current language. As a pattern classification problem, speech recognition must be mathematically formulated and decomposed into simpler subproblems.

Let X be a sequence of feature vectors obtained from the speech signal

$$X = X_1, X_2, X_3, ..., X_m$$

the feature vectors X_i are generated sequentially by increasing values of i and m is the number of feature vector in the sequence.

Let W be the word content of the speech signal

$$W = w_1, w_2, w_3, ..., w_n$$

where n is the number of words in the speech signal.

$P(W|X)$ is the probability that the words W were spoken, given the feature vector sequence, which is called "observation". After defining these elements, the speech recognition can be defined as a decision making process searching for the most probable word sequence $\hat{W}$

$$\hat{W} = arg \max_{W} P(W|X) \tag{2.3}$$

which consists of searching for the most likely word sequence W conditioned on observation sequence X. The probability $P(W|X)$ cannot be observed directly because of randomness of feature vector space. We need to rewrite this probability.

The right-hand side probability of equation (2.3) can be rewritten according to Bayes' formula of probability theory as

$$P(W|X) = \frac{P(W)\ P(X|W)}{P(X)} \tag{2.4}$$

where $P(W)$ is the prior probability that the word string W will be spoken by the speaker, $P(X|W)$ is the likelihood, and $P(X)$ is the average probability that X will be observed. $P(X)$ in the equation (2.4) is known also as evidence, and it is generally omitted in speech recognition since this probability is same for all acoustic signal observations. The new version of maximization equation (2.3) can be rewritten as

$$\hat{W} = arg \max_{W} P(W)\ P(X|W) \tag{2.5}$$

after omitting the evidence of observing acoustic observation X in Bayes' formula.

Equation 2.5 is the base for classification in speech recognition. By writing the equation in this form we have the opportunity of computing the probabilities $P(W)$ and $P(X|W)$ by training some models. $P(W)$ can be obtained by training a model for the language and is independent of acoustic information. **Language modeling** is based on assigning a probability to each word occurrence within a context and the model can be trained on a large text containing virtually all occurrences of word sequences in the language.

The second probability in equation (2.5) can be obtained by training a model for the acoustic realizations of words. This modeling is called **acoustic modeling** and can be obtained by training a model from a large acoustic database which contains virtually all realizations of the words in the language.

The "classification" process shown in Figure 2.5 recieves inputs from two models: acoustic and language models. The following sections explain in detail these models and modeling techniques used to obtain them. Before explaining the modeling techniques we must give some definitions related to speech recognition.

Utterance is the vocalization (speaking) of a word or words that represent a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences.

Speaker dependence, is the dependency of speech recognizer to a specific speaker. Speaker dependent recognizers are generally more accurate for the correct speaker, but much less accurate for other speakers. Speaker independent recognizers are designed for a variety of speakers. Adaptive systems usually start as speaker independent systems and utilize training techniques to adapt to the speaker to increase their recognition accuracy.

Vocabulary is the list of words or utterances that can be recognized by the recognizer. Generally, smaller vocabularies are easier for a computer to recognize, while larger vocabularies are more difficult. The entries can be multiple-words, words or sub-word units. Vocabulary size can vary from two words (e.g. "yes" and "no") to several thousand words.

Accuracy is the performance measure for the recognizer. This includes not only correctly identifying an utterance but also identifying if the spoken utterance is not in its vocabulary. Good ASR systems have an accuracy of 98% or more. The acceptable accuracy of a system really depends on the application.

Training and **testing** are the main steps of recognizer. Training is determination of model parameters and testing is using the models to recognize a speech signal. Testing is called also as recognition. Training can be considered as configuration of recognizer before it can be used.

2.8 Acoustic Modeling

Acoustic modeling is the process of generating models for each class in speech recognition. The class can be a word, a sub-word unit or a phoneme. There are many kinds of acoustic models and modeling techniques. The simplest acoustic model can be, for example, the acoustic realization of each words in the vocabulary of speech recognizer.

Figure 2.8 give the acoustic modeling process. Acoustic modeling process is not a part of speech recognition. It provides the acoustic models which are used in speech recognition for classification task as shown on Figure 2.5.

The flowchart in Figure 2.8 is not standard for all acoustic modeling techniques but it includes the common steps in acoustic modeling.

The first step is "initialization" of models. At this step pre-segmented feature vectors are assigned to classes and a model for each class is created.

In "training" step initial models are used for classification of new feature vectors which are not segmented. After segmentation the new class boundaries for models are determined in an iterative approach. Some generalization algorithms are applied to have a better modeling of unseen data. The output of this process is acoustic models for each class. The acoustic models are used by the recognizer to determine the probability $P(X|W)$ of equation (2.5).

An important aspect of classification process is distance measure which is common in training of acoustic models and later testing. All acoustic modeling techniques are based on some distance measures to test the closeness of a new feature vector to a model. Distance

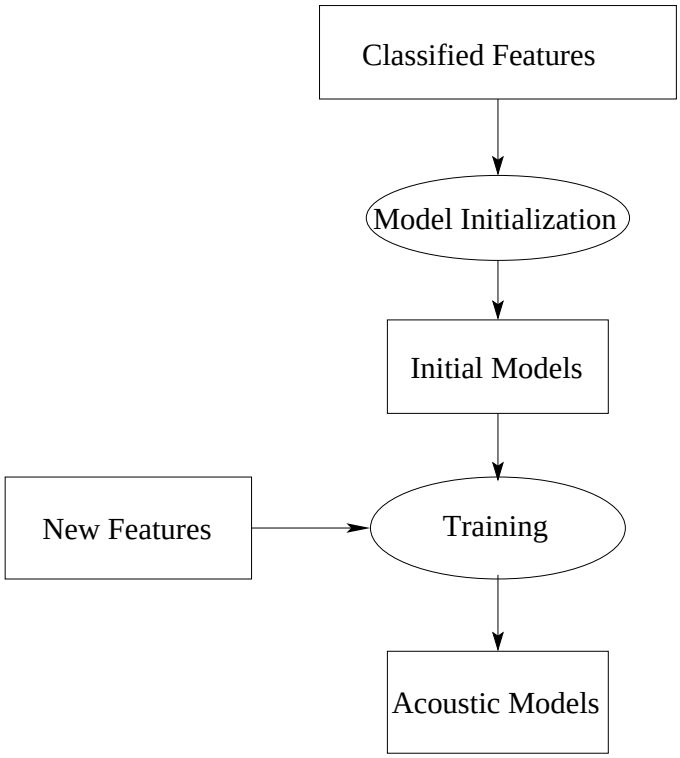

Figure 2.8: Acoustic modeling

measure is used for comparing feature vectors to some stored templates for classification purposes. The stored templates can be updated with new data in an iterative approach.

Let C be the available classes represented by a template feature vector.

$$C = c_1, c_2, ..., c_N \tag{2.6}$$

N is the number of classes. The simplest way to classify a feature vector x_i is to compare it to class templates and find the closest template.

$$J = arg \min_{j=1}^{N} d(x_i, c_j) \tag{2.7}$$

where J is the class for feature vector x_i and $d(\ ,\ )$ is the distance function.

The most commonly used distance function is Euclidean distance function which is defined as:

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \tag{2.8}$$

for vectors x and y.

The main acoustic modeling techniques are:

1. Dynamic Time Warping

2. Vector Quantization
3. Hidden Markov Modeling (HMM)
4. Artificial Neural Network (ANN) Modeling
5. HMM/ANN Hybrid Modeling

The first two techniques are based on template matching techniques and can be combined. The last three techniques are statistical acoustic modeling techniques and are based on class probability density functions. The following subsections will give some detailed description of each method.

Todays' speech recognizers, in general, use "hidden Markov model" based acoustic modeling. Recently, use of hybrid HMM/ANN model based acoustic modeling is increased due to better performances obtained [10].

2.8.1 Dynamic Time Warping Based Models

Dynamic time warping (DTW) is based on "dynamic programming" which is defined as "an algorithmic technique in which an optimization problem is solved by caching subproblem solutions rather than recomputing them". It includes a backtracking process which guarantees the best solution for a problem.

DTW is a template matching method for classification of patterns (acoustic vectors in speech recognition). DTW based acoustic modeling includes creation of templates for each class. Templates can include the feature vectors of a pre-recorded word to be recognized in case of word recognition or can include mean feature vectors linked to some phonemes if the task is phone recognition.

Two pronunciations of same word in different times are not the same because of speed and duration changes in the speech. This nature of speech needs time alignment when it is being compared to a pre-recorded version which has the same phonetic content. DTW is one of the techniques to make this time alignment. Figure 2.9 shows an example of time-alignment and the application of DTW algorithm to compare two realizations of same word.

DTW tries to find the minimum distance between reference and test templates. The computation of minimum distance is based on the following formula:

$$D(i,j) = d(i,j) + min \left\{ \begin{array}{c} D(i,j) \\ D(i,j-1) \\ D(i-1,j) \end{array} \right\}, \; i = 1,n; \; j = 1,N. \tag{2.9}$$

In equation (2.9) the indexes i and j are for the feature vectors being compared, $d(i,j)$ is the Euclidean distance between two feature vectors and $D(\ ,\ )$ is the global distance until

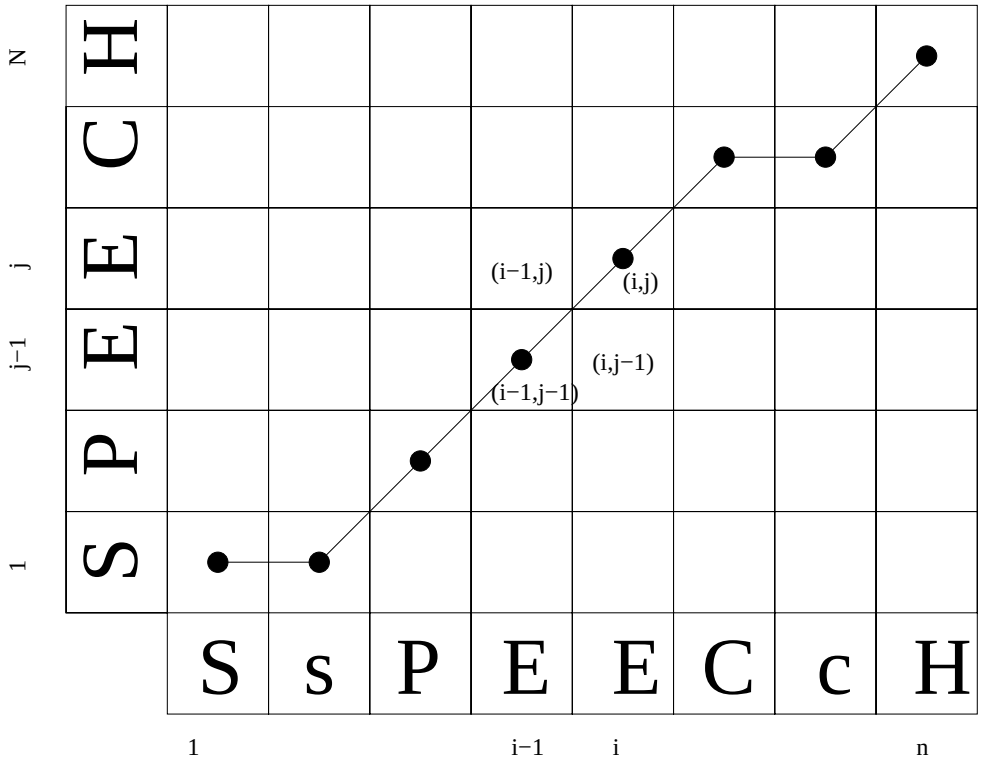

Figure 2.9: Dynamic time warping. N is the number of frames in reference template, n is the number of frames in the pattern to be matched.

the current point. At the starting point the distance is:

$$D(0,0) = d(0,0)$$

The global distance between a reference template of length N and the observed test sequence of length n is $D(N,n)$ and can be calculated recursively.

Recognition of a observed test sequence X with a DTW based speech recognizer which has M words in its vocabulary W, can be defined as a search problem as follows:

$$k = arg \min_{l=1}^{M} D(N_l, n) \tag{2.10}$$
$$\hat{w} = W_k$$

where k is the index of the word in the vocabulary closest to the observed speech data. N_l is the number of feature vector in l^{th} word template and n is the number of feature vector in observation data. $\hat{w}$ is the recognized word.

The recursive version of DTW algorithm can be very slow due to high number of feature vectors available even for small words. There are 100 frames in 1 second of speech (10 ms frame shift). Fortunately there is a method which require less memory and quicker than the recursive version. This method needs two arrays of length N_l (the length of reference

template). These two arrays keep the previous and current distances at any time i which are sufficient to compute the distances of equation (2.9). The arrays are slided with increasing i.

2.8.2 Vector Quantization Models

Vector Quantization (VQ) is the process where a continuous signal is approximated to a digital representation (quantization) considering a set of parameters to model a complete data pattern (vector). VQ helps reducing the feature dimension and this leads to computational efficiency. VQ is generally used in speech coding [11].

Vector quantizer is defined as a system for mapping a sequence of continuous or discrete vectors into a digital sequence suitable for digital communication over a digital channel [12]. The principle goal is data compression.

Vector quantization can be seen as a clustering problem. Since the feature space is continuous and it is computationally expensive to work in a such space, clustering provides a number of discrete and finite vector (codebook) which can be used instead of real feature vectors. These codebooks are similar vectors to feature vectors and are the center of clusters. After defining VQ as a clustering problem, acoustic modeling is the determining of cluster boundaries and assigning clusters to pre-defined acoustic classes. The details of VQ methods and algorithms can be found in [12].

VQ models include a number of codebook obtained after a training process. The clusters represented by codebook vectors are associated to phonetic classes. Phonetic classification of speech feature vectors can be done by labeling each feature vector by the index of nearest codebook vector. Speech recognition using VQ models can be summarized as follows:

1. Obtaining feature vectors for training and test speech data.
2. Training VQ codebooks using a phonetically labeled training data.
3. Assigning VQ codebooks to phonetic classes using phonetic labels from training data.
4. Mapping feature vectors of test data to VQ clusters using Euclidean distance measure.
5. Labeling test data with phonetic labels associated to VQ codebooks.
6. Finding the best fitting word which corresponds to the phonetic representation of test data.

The main advantages that VQ representations give for speech technology consider the following points:

- Efficient storage of relevant information. The reduction of the dimensionality of speech patterns and training sets length, is very important for optimizing the storage and transmission of signals.

- Reduced training sets generation. The generation of reduced codebooks for representing speech information has a direct implication in fast response retrieval and memory constraints of speech recognizers. VQ representations optimize the determination of similarities between a pair of vectors by means of simple lookup tables of codewords.

- Simplified labeling process. The process of labeling can be easily done using VQ techniques. The association of a phonetic class directly to each codeword can be equivalent to assigning a phonetic label to each speech frame.

The main disadvantages of using VQ representations can be stated as follows:

- Distortion. VQ coding of information, carry an implicit distortion of the original signal due to quantization constraints. As the number of bits used to represent a pattern decrease, the quantization error increases.

- Codebook Storage. The selection of an adequate number of codewords is often a difficult trade-off to establish between accuracy and speed-storage constraints.

VQ based speech recognition works well for small vocabulary tasks, [13], [14]. It is possible to use VQ modeling techniques with other acoustic modeling techniques for better recognition accuracies. When combined with DTW, instead of computing distances for all feature vectors in DTW process, distances of VQ codebooks can be computed at the beginning and can be used directly during DTW process. This can reduce computational complexity of DTW based speech recognition.

VQ can also be used as a front end acoustic processor to hidden Markov modeling based speech recognition systems.

2.8.3 Hidden Markov Models (HMM)

Hidden Markov modeling (HMM), unlike the previously defined two other modeling techniques, is a statistical modeling technique. Statistical modeling tries to characterize statistical properties of a signal. Statistical modeling is based on assumption that signal can be characterized as a parametric random process, and the parameters of the stochastic process can be determined in a well-defined manner.

In HMM, speech is assumed to be a discrete stationary process. Every acoustic utterance is modeled as a series of discrete stationary states with instantaneous transitions between them. The states are organized to obtain a chain that model the entire sentences. The details of HMM can be found on [5].

Each word or word sequence is defined as a state chain which is called as model. By replacing the word sequence (W), in the equation (2.5) with the HMM representation of the word sequence (M), we can rewrite the speech recognition problem as:

$$\hat{M} = arg \max_{M} \; P(M) \; P(X|M) \tag{2.11}$$

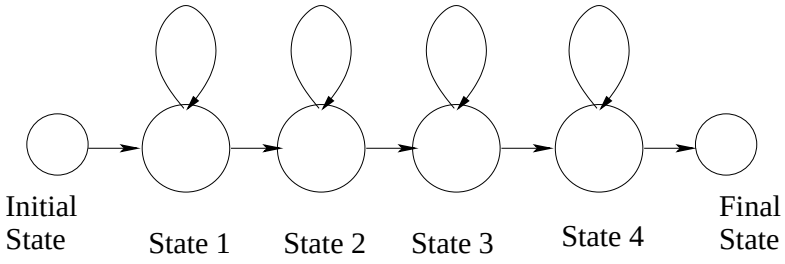

Figure 2.10: Hidden markov model (HMM).

finding the model that produced the observation, $P(X|M)$, is equivalent as finding the word sequence represented by the observed acoustic signal. $P(M)$ is the probability that word sequence can be observed and it is obtained from language model.

A typical HMM is shown on Figure 2.10. The state sequence is hidden. Only the observations are available to determine the parameters of HMM. There are two types of probabilities in an HMM; *transition probabilities* and *emission probabilities* of each state. Observations are the result of a stochastic process associated with transition probabilities and emission probabilities of states.

There are two method to compute the probability $P(X|M)$.

- Maximum likelihood criterion
- Viterbi criterion

Maximum likelihood criterion method finds the maximum emission probability of an HMM for a given observation sequence by using Baum-Welch algorithm.

Viterbi criterion considers only the best path through M. This method is more efficient than maximum likelihood method and is based on a similar type of pruning like DTW. More detailed information on these probability reestimation methods can be found on [15], [5] and [6].

In training phase a pre-classified and labeled training database is used to estimate HMM transition probabilities and the parameters of output probability density functions for each HMM.

The base speech unit for an HMM based acoustic modeling can be various. It is possible to start with word HMMs and by combining them sentence HMMs are created. The probability is estimated on a sentence level. This type of modeling can be used for speech recognition systems which are capable to recognize only few sentences with a limited word number in the vocabulary since it is difficult to train the models with sufficient amount of data. A recognizer of this type is called as small vocabulary speech recognizer.

It is also possible to have a large number of words in the dictionary when the speech units are smaller than words. In the case of phonetic acoustic models, it is possible to increase the number of words to be recognized with speech recognizer without increasing the amount of

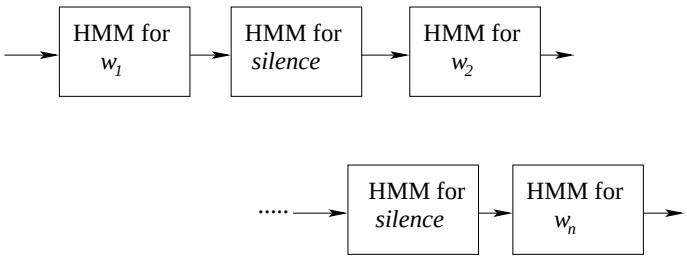

Figure 2.11: HMM concatenation to create sentence HMMs from word HMMs. The sentence W is defined as, $W = w_1, w_2, ..., w_n$

training data needed to train the HMMs. There is a limited number of phonemes, less than 100, to model and virtually it is sufficient to have several instances of each phoneme in the training database (practically the need for training data is higher since the context of the phoneme is also important to have a good model of it).

A phoneme based HMM is constructed as follows [16]:

1. Create a phonetic dictionary which include phonetic transcriptions of each words in the vocabulary.
2. Define an elementary HMM for each phoneme with distinguished starting and final states.
3. The HMM of a word is created by concatenating the HMMs of each phoneme in the phonetic transcription of the word.
4. The HMM of a sentence is created by adding a silence HMM between the words of the sentence as shown in Figure 2.11.
5. Using a parameter estimation algorithm (Baum-Welch or Viterbi) the parameters of HMM are determined from a sufficiently large training database.

The recognition of a test utterance is a search problem. The word sequence represented by the most likely HMM that have created the test data (the HMM with highest probability) is selected as the output of the recognizer.

2.8.4 Neural Network Based Models

Neural network is defined as "a software that is *trained* by presenting it examples of input and the corresponding desired output". Neural networks are general-purpose programs which can be used in almost any problem that can be regarded as pattern recognition.

Neural networks simulate the parallel architecture of animal brains. They have the following properties:

- They are composed of simple processing elements called "neuron".

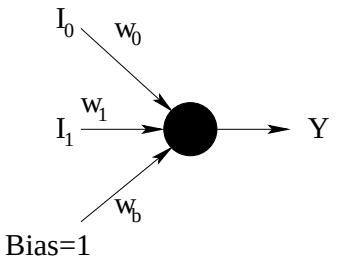

Figure 2.12: The structure of a single neuron.

- There is a high degree of interconnection between neurons.
- The messages transfered between neurons are simple scalar messages.
- The interactions between neurons are adaptive.

A biological neuron can have thousands of inputs and a lot of output but the connections of artificial neurons are limited with the computational power of computers.

A sample neuron is shown on Figure 2.12. This neuron have two inputs, one output and one bias unit. Training of this neuron consists in updating the weights to obtain the desired output for specified inputs. The neuron can be designed as a comparator. For example:

$Y = 1$ if $(w_0 * I_0 + w_1 * I_1 + w_b) > 0$

$Y = 0$ if $(w_0 * I_0 + w_1 * I_1 + w_b) \leq 0$

Determining the weights, or "training" the network, is essentially done by minimizing some error functions. Error function is commonly selected as "sum (over all the output nodes and all the examples in the training data set) of squares of the differences between actual and desired output node values".

The adaptation procedure of a neural network can be defined as; "change the weight by an amount proportional to the difference between desired output and the actual output". This method is called *Perceptron learning rule* and is formulated as:

$$\Delta w = \eta * (D - Y) * I_i \tag{2.12}$$

where Δw is the weight change, D is the desired output, Y is the actual output and I_i is the i^{th} input.

Perceptron is a simple neuron. It computes a single output from multiple inputs by forming a linear combination according to the weights. A single perceptron is not very useful because of its limited mapping ability. But it can be used as a building block for larger neural networks called *Multi-layer perceptron(MLP)*. It is called multi-layer because of organization of neurons in a layered architecture. There is an input layer, a hidden layer and an output layer in MLP. The neurons of each layer have different processing functions. MLPs are capable of approximating any continuous function to any given accuracy with a sufficiently large hidden layer [17].

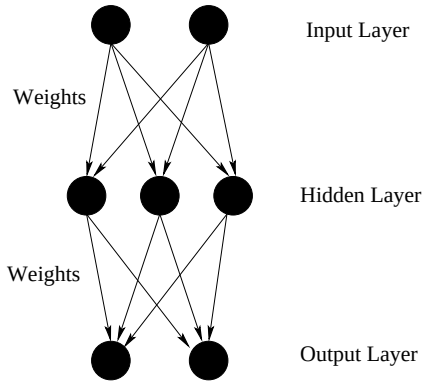

Figure 2.13: The structure of MLP and connections between neurons.

Figure 2.13 shows a sample MLP structure. As can be seen on the figure, MLP is a feed-forward network, the message flow is from input to output.

Although MLPs have no feedback, the training process does have feedback; the output node values are used to change the weights, and through the weights all the nodal values. MLP networks are trained using supervised learning which needs a pre-classified training data.

In speech recognition, MLP can be trained directly by the feature vectors or it is possible to use VQ techniques to have a limited number of different feature vectors at the input. It is possible to use a context in training of MLP, that means when training the MLP with a feature vector, it is possible to give the previous n feature vectors and the next n vectors as input to the MLP. When the context is used for training MLP, the number of inputs is augmented which means the computational load is increased. But use of context is important since it is the best way to model transitions between speech units. A good reference for use of neural networks for speech recognition is [18].

MLP based speech recognition commonly use phonemes as base speech units to model. The outputs are phonemes or small speech units. Training procedure of MLP networks for speech recognition can be summarized as follows:

- Define context to be used at the input. For example 5 previous and 5 next frames which makes 11 feature vectors at the input. If the feature vectors have a size of 13, then the input layer have 143 neurons.
- Define number of neurons on hidden layer. A large hidden layer gives better accuracies but increases computational load.
- Define output layer neurons. Generally one neuron for each phoneme.
- Train the neural network to have minimum classification error for a pre-classified training data set.

After having obtained the neural network, it can be used for classification of new speech data. The output obtained during this classification process is a set of probability for each output which called "posterior probability". This probability can be used then by, for example, Viterbi decoding algorithm to find the best matching word sequence for the current test data. When the outputs of the MLP are the phoneme posterior probabilities, phonetic transcriptions of words are needed by Viterbi decoding process to find the matching word.

Recently combination of MLP networks with HMMs resulted good recognition performances. This type of use is the subject of next subsection.

2.8.5 HMM/ANN Hybrid Models

HMMs are widely used in acoustic modeling for speech recognition. HMMs provide a good representation of sequential nature of speech and they benefit from powerful training and decoding algorithm. HMMs have some disadvantages which can be avoided when combined by "artificial neural networks" (ANN). These disadvantages generally come from optimizations for parameter estimations and they are:

1. Insufficient discriminative ability due to maximum likelihood technique which tries to maximize the likelihoods of state sequences instead of maximizing posterior probabilities.
2. Assumptions about the statistical distributions of states.
3. Assumptions about the topology of states and use of first order Markov chains.
4. Absence of acoustic context.

The Neural networks are used efficiently in speech recognition [19]. Efficient learning and classification abilities make neural networks powerful on pattern recognition problems like speech recognition. MLP networks are commonly used neural networks for speech recognition. They have several advantages:

1. Discriminative learning ability.
2. Modeling capability of any kind of nonlinear functions of input thanks to hidden layer.
3. Flexible architecture and possibility to use context information.
4. Possibility of hardware implementation with parallel computers.

ANNs also have disadvantages when used in speech recognition. They have limitations on classifications time sequential input data like speech since they are static pattern classifiers and dynamic nature of continuous speech is not suitable as input for ANNs. There are successful implementations of ANN based speech recognizer for isolated word speech recognition tasks[19], [20], [21].

The combined (hybrid) HMM/ANN acoustic modeling techniques gives better results than the use of one of them only[18]. This combination take advantage of good classification performance of ANN and good sequential modeling property of HMM. MLP networks are used to estimate HMM emission probabilities.

In hybrid HMM/MLP approach the maximum likelihood estimation of $P(X|M)$ in equation (2.11) of HMM emission probabilities is replaced by maximum a posteriori (MAP) estimation, $P(X|M)$, which is provided by MLP. The MAP probability of HMM M_i is defined as:

$$P(M_i|X) = \sum_{l=1}^{L} P(q_l^n, M_i|X), \forall n \in [1, N] \tag{2.13}$$

where N is the number of frames, q_l^n means the HMM state q_l is visited at time n, X is the observation sequence and M_i is the HMM model. From the Bayes' formula of equation (2.4) we can see that these two probabilities are equivalent since $P(M)$, the language model probability, and the $P(X)$, the a priori probability of observing acoustic sequence X, are not used during acoustic modeling. By using MAP instead of maximum likelihood estimation the discrimination power of HMMs is improved.

HMM/MLP hybrid speech recognizers can be obtained in three steps:

1. Network specification.
2. Training
3. Recognition

Network specification includes use of feed-forward MLP networks with three layer:

- Input layer include several consecutive frames, for example 9, to have context information at the input. Use of continuous feature vectors instead of using VQ codebooks, is desired for better performance.
- Hidden layer includes a number of neurons, for example 1000, with sigmoidal activation function.
- Output layer includes the HMM states. In case of using one state phone models, the number of neurons in the output layer is the number of phonemes, for example 50.

Training includes selection of training methods and determining training procedure. The common weight update algorithm is error back-propagation [18], [22]. Random sampling of training database provides rapid convergence without using all training samples. Cross-validation is a way of testing the performance. The use of a nonlinear function like standard sigmoid function in neurons minimizes classification errors when compared with linear activation functions. Training phase of an HMM/MLP includes re-segmentation of input data

once the optimum MLP is obtained. After re-segmentation, training is restarted, after several iteration, the MLP with best cross-validation score is selected for recognition.

Recognition includes use of HMM decoding process. MLP is used to generate posterior probabilities given feature vectors with their context. Viterbi decoding is then applied and the most probable word sequence is obtained.

2.9 Language Modeling

Language modeling is the process of extracting important properties of a natural language by analyzing statistically a corpus of language. The goal is to assign probabilities to strings of words in the language. These probabilities are then used to rank the word sequence candidates from recognition results of acoustic model. As stated in section 2.7, probability that the word sequence W were spoken given the feature vector X, $P(W|X)$, can be rewritten from Bayes' formula as:

$$P(W|X) = \frac{P(W)\ P(X|W)}{P(X)} \tag{2.14}$$

where $P(W)$ is the probability that the word string W will be spoken by the speaker, $P(X|W)$ is the probability that when the speaker says W the speech signal represented by X will be observed, and $P(X)$ is the average probability of observing X.

In equation (2.14) the probability $P(X|W)$ is the acoustic model probability which was the subject of the previous section, $P(X)$ is omitted because of assumption about randomness of speech. The last unknown probability is the probability $P(W)$, the language model probability, is the subject of this section.

By using a language model for speech recognition, the number of acceptable word sequences is limited. This limitation leads to an increase in the accuracy of the speech recognizer since some erroneous word sequences will be replaced by nearest approximations which are mostly the correct word sequences. Language models are useful for large vocabulary continuous speech recognition tasks. For small vocabulary isolated word speech recognition tasks there is no need for language models since each utterance is composed of only one word.

For small vocabulary connected word speech recognition the language models can be deterministic and can be obtained by creating finite state automata for word sequences that can be seen.

In this chapter only statistical language models are considered since the other language model types like finite state automata can be integrated in the decoding algorithm of speech recognition and they don't require a special training session. This type of language model is application dependent.

Language models are used to assign probabilities to word sequences. Models are trained with a large text corpus from the language to be modeled. Language modeling is based on estimation of probability that word sequence W can exists in the language. For a word

sequence $W = w_1, w_2, ..., w_N$, probability, $P(W)$ is defined as:

$$P(W) = \prod_{i=1}^{N} P(w_i|w_1, w_2, ..., w_{i-1}) \tag{2.15}$$

where N is the number of words in the sequence. $P(w_i|w_1, w_2, ..., w_{i-1})$ is the probability that w_i is observed after word sequence $\{w_1, w_2, ..., w_{i-1}\}$ which is called *history*.

Statistical language modeling is based on the formulation in equation (2.15). The main task in language modeling is to provide good estimations of $P(w_i|w_1, ..., w_{i-1})$, the probability of i^{th} word given the history $\{w_1, ..., w_{i-1}\}$ [16]. There is two methods frequently used for language modeling:

- *n-gram* language models
- *part-of-speech* (POS) based language models

The details of these two types of modeling techniques will be explained in following sub-sections. Both of them are based on statistics obtained from a training corpus. *n-gram* language models are based directly on the occurrences of words in the history list where POS models use linguistic informations instead of words.

Language modeling is based on counting the occurrences of word sequences. When long histories are used some word sequences may not be appear in the training text. This results in poor modeling of acceptable word sequences and is called as data sparseness problem.

Sparseness problem in language modeling is solved by applying smoothing techniques to language models. Smoothing techniques are used for better estimating probabilities when there is insufficient examples of some word sequences to estimate accurate word sequence probabilities directly from data. Since smoothing techniques are applied to *n-gram* language modelings, some of smoothing techniques will be presented in following sub-section.

When the number of possible word sequences that can be accepted by the speech recognizer is known and is limited, then it is possible to create some finite state grammars which limits the output of the recognizers. The finite state grammars used in this case are also called language models. This type of language models are task oriented and can be created in a deterministic way.

2.9.1 N-gram Language Models

N-gram language models are most widely used language modeling techniques. The N is selected as 1 (unigram), 2 (bigram) or 3 (trigram) in most *n-gram* language models.

As stated earlier, $P(W)$ is the the probability of observing word sequence W and can be decomposed as:

$$P(W) = P(w_1, w_2, ..., w_n)$$

$$
\begin{aligned}
&= P(w_1)P(w_2|w_1)P(w_3|w_1,w_2)\ ...\ P(w_n|w1,w_2,...,w_{n-1}) \\
&= \prod_{i=1}^{N} P(w_i|w_1,w_2,...,w_{i-1}) \qquad (2.16)
\end{aligned}
$$

where $P(w_i|w1,w_2,...,w_{i-1})$ is the probability that w_i will be observed after history $w_1,w_2,...,w_{i-1}$. This formulation is the general form for *n-gram* language models. For unigram language model the probabilities $P(w_i)$, for bigram $P(w_i|w_{i-1})$ and for trigram $P(w_i|w_{i-1},w_{i-2})$ are computed.

The size of history depends on the selection of n for an *n-gram* language. There is no history when for unigram language models, the history has only one word for bigram and two words for trigram language models.

The probability $P(W)$ is computed by counting the frequencies of word sequence W and the history. For example trigram probabilities are computed as:

$$P(w_i|w_{i-2},w_{i-1}) = \frac{C(w_{i-2},w_{i-1},w_i)}{C(w_{i-2},w_{i-1})} \qquad (2.17)$$

where $C(w_{i-2},w_{i-1},w_i)$ is the number of occurrences of word sequence w_{i-2},w_{i-1},w_i and $C(w_{i-2},w_{i-1})$ is the number of occurrences of history w_{i-2},w_{i-1}.

In order to have a good estimate of language model probabilities we need a large text corpus including virtually all occurrences of all word sequences. For trigrams a corpus of several millions of words can be sufficient but for higher values of n the number of words should be very high.

Perplexity

The efficiency of an *n-gram* language model can be simply evaluated by using it in a speech recognition task. Alternatively it is possible to measure the efficiency of a language model by its *perplexity*. Perplexity is a statistically weighted word branching measure on a test set. If the language model perplexity is higher, the speech recognizer needs to consider more branches which means there will be a decrease on its performance.

Computation of perplexity does not involve speech recognition. It is defined as the derivative of cross-entropy [23]. The perplexity based on cross-entropy is defined as:

$$PP(W) = 2^{H(W)} \qquad (2.18)$$

where $H(W)$ is the cross-entropy of the word sequence W and is defined as:

$$H(W) = -\frac{1}{N}\log_2 P(W) \qquad (2.19)$$

where N is the length of word sequence and $P(W)$ is the probability of the word sequence from language model. It must be noted that W is a sufficiently long word sequence which helps to find a good estimate of perplexity.

Perplexity can be measured for the training set and the test set [23]. When it is measured for training set it provides a measure of how well the language model fits the training data, for the test set it gives a measure of the generalization capability of language model. Perplexity is seen as a measure of performance since it correlates with better recognition results. Higher perplexity means there will be more branches to consider statistically for a recognition task which leads to lower recognition accuracies.

Smoothing

Another important issue in *n-gram* language modeling is *smoothing*. Smoothing is defined as adjusting the maximum likelihood probabilities, obtained by counting to model word sequences, to produce more accurate probability distributions. This is necessary since data sparseness problem in training data due to high number of available word sequence may result in assigning low probabilities or zeroes to certain word sequences that will probably seen in test data. The purpose of smoothing is to make the probability distributions more uniform which means assigning higher probabilities to word sequences with low probabilities obtained by counting, and assigning low probabilities to word sequences with too high probabilities. This gives better generalization capability to the language model.

A good smoothing example is; to consider each bigram is occurred one more time than it occurred in the training set. Which can be done as:

$$P(w_i|w_{i-1}) = \frac{1 + C(w_{i-1}, w_i)}{\sum_{w_i}(1 + C(w_{i-1}, w_i)} \tag{2.20}$$

by modifying equation (2.17). By doing such a simple smoothing we avoided zero probabilities which could be harmful to the speech recognizer since it can reject a correct word sequence that could not appeared in training set of language model but had a higher probability from acoustic model.

There are several smoothing techniques that can be used for language models. For different smoothing techniques [23] is a good reference. We will consider only the **back-off smoothing** (Katz back-off model) technique which is commonly used.

Katz back-off smoothing is based on Good-Turing estimates which partition *n-grams* into groups depending on their frequency of appearance in the training set. In this approach the frequency,r, of an *n-gram*, n is replaced by r_* which is defined as:

$$r_* = (r + 1)\frac{n_{r+1}}{n_r} \tag{2.21}$$

where n_r is the number of *n-grams* that occurs exactly r times and n_{r+1} is the number of *n-grams* that occurs exactly $n + 1$ times. The probability of an *n-gram*, a, is then defined as:

$$P(a) = \frac{r_*}{N} \tag{2.22}$$

where N is the number of all counts in the distribution. In Katz smoothing, the *n-grams* are partitioned into three class according to their frequencies in the training set. For partitioning a constant count number, k, is used. This is a predefined number generally selected between 5 and 8. If r is the count of an *n-gram*:

- Large counts are considered as reliable and there is no smoothing; $r > k$.
- The counts between zero and k are smoothed with Good-Turing estimates; $0 < r \leq k$. This smoothing is a *discounting* process which use a ratio based on Good-Turing estimate to reduce the lower counts.
- The zero counts are smoothed according to some function, α, which tries to equalize the *discounting* of nonzero counts with increasing zero counts by a certain amount.

For a bigram language model, the Katz smoothing can be summarized as follows [23] [24]:

$$P_{Katz}(w_i|w_{i-1} = \begin{cases} C(w_{i-1}w_i)/C(w_{i-1}) & if \quad r > k \\ d_r \; C(w_{i-1}w_i)/C(w_{i-1}) & if \quad k \geq r > 0 \\ \alpha(w_{i-1})P(w_i) & if \quad r = 0 \end{cases} \tag{2.23}$$

where;

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \tag{2.24}$$

and,

$$\alpha(w_{i-1} = \frac{1 - \sum_{w_i;r>0} P_{Katz}(w_i|w_{i-1})}{1 - \sum_{w_i;r>0} P(w_i)} \tag{2.25}$$

It can be seen from equation (2.25) that the probability of zero count bigrams is increased by weighing unigram probabilities with α.

There are several disadvantages of *n-gram* language models:

- They are unable to incorporate long-distance word order constraints since the length of history is generally small and the exact order is considered.
- They are not suitable for flexible word order languages like Turkish.
- It is not possible to integrate new vocabulary words or alternative domains into language models.
- The meaning cannot be modeled by *n-gram* language models.

Despite these disadvantages, *n-gram* language models gives good results when used in speech recognition tasks because they are based on a large corpus with helps to model the approximate word orders that exist in the language. Many languages have a strong tendency toward standard word order.

Some of the disadvantages of *n-gram* language models can be avoided by using clustering techniques. For example, use of meta-words for some class of words like days of week, months, cities, person names.

Clustering can be made manually or automatically on training set. Clustering can improve the efficiency of language model by creating more flexible models. The next subsection gives details of a clustering technique, *part-of-speech* (POS) tagging.

2.9.2 POS Modeling

Part-of-speech (POS) tagging based language modeling is one of the class based modeling techniques which try to alleviate data sparseness problem of language models.

Class based language models are based on classifying the words that have same behavior in the sentence, to one class. The words in same class are then replaced by the class identifier when they occurred in the training data. A good word classification can decrease the number of different *n-grams* available in the language model which improves the search performance, and can model unseen word sequences which will improve recognition performance of speech recognizer. Classes can be, for example, the days of week, the months, the seasons, the proper names, etc.

Class based language modeling is generally based on classes defined from semantic or grammatical behavior of words. These types of class based language models are claimed to be effective for rapid adaptation and reduced memory requirements [23].

The *n-gram* probability of equation (2.15) can be adapted to class based language modeling by introducing a class history.

$$P(w_i|c_{i-n+1},...,c_{i-1}) = P(w_i|c_i)P(c_i|c_{i-n+1},...,c_{i-1}) \tag{2.26}$$

From equation (2.26) we can see that the probability of observing the word w_i depends on the class history $c_{i-n+1},...,c_{i-1}$ where n is the degree of *n-gram*, $P(w_i|c_i)$ is the probability that word w_i belongs to the class c_i, $P(c_i|c_{i-n+1},...,c_{i-1})$ is the probability of class c_i given the class history. From this equation we can rewrite trigram probabilities as:

$$P(W) = \sum_{c_1,...,c_n} \prod P(w_i|c_i)P(c_i|c_{i-2},c_{i-1}) \tag{2.27}$$

where the mapping of words to classes can be many to many, that means a word can be assigned to more than one class and a class can contain more than one word. If there is a restriction on this mapping which restricts the assignment of each word to only one class, the equation (2.27) can be rewritten as:

$$P(W) = \prod P(w_i|c_i)P(c_i|c_{i-2},c_{i-1}) \tag{2.28}$$

The classification in POS language modeling is based on affixation behavior, abstract semantic topologies, historical development etc. A typical example of POS categories could be *noun, verb, adjective, adverb, pronoun, preposition, determiner, conjunction, interjection.*

POS classification consists in assigning POS tags to words according to their usage in a context. The classification procedure can be automatic which can be trained by a pre-tagged text. Corpus based statistical POS taggers are generally based on hidden Markov modeling techniques, statistical decision tree techniques or transformation based learning.

In [26] it is claimed that only a lexicon and some unlabeled training text is sufficient to obtain an accurate tagger.

As shown in [25], the maximum entropy based tagger achieves state-of-art parsing accuracies.

A decision tree based POS tagger approach is shown in [28]. It is shown that a decision tree based tagger is as accurate as other methods and is easy to integrate the resulting tagger to speech recognition as well as natural language processing.

Recently a tagger named TnT [27] is claimed to give good results after application of an appropriate smoothing.

In chapter 4, a different approach for class based language models will be given for Turkish language modeling which is a free constitute and agglutinative language. This approach can replace the POS based modeling techniques given above for agglutinative languages.

2.10 Classification of Speech Recognizers

Speech recognizers are classified according to:

1. Size of their vocabulary: small vocabulary, large vocabulary.
2. Dependence to a particular speaker: Speaker dependent, speaker independent.
3. Continuousness of speech that they recognize: Isolated, connected, continuous.
4. Keyword spotting capability.

The classification of speech recognizers is used for determining the complexity of speech recognition task accomplished by speech recognizer. The complexity level of speech recognizers is given on Table 2.4 [29].

Speech recognizers that have highest complexity are large vocabulary continuous speech recognizer. This type of recognizers are the ultimate goal of speech recognition. The other speech recognizers are the sub-classes of this speech recognizers. High complexity of large vocabulary speech recognition tasks leads to less accurate speech recognizers which cannot be used in practical applications. Defining sub-class of speech recognizer helps obtaining more accurate and application oriented speech recognizer which can be used in real life such as commercial applications.

As it can be seen from classification criteria, the classification is fuzzy. For example, the size of vocabulary can be different to determine if a certain speech recognizer is small or large

Table 2.4: Classification of speech recognition systems and the complexities. "1" is for least complex, "10" is for most complex speech recognition system.

	Isolated		Connected		Continuous	
Speaker	small	1	small	4	small	5
Dependent	large	4	large	5	large	6
Multi	small	2	small	4	small	6
Speaker	large	4	large	5	large	7
Speaker	small	3	small	4	small	5
Independent	large	5	large	8	large	10

vocabulary speech recognizer. With increasing computing power of todays' technology and improved speech recognition algorithms, the limits can be modified. To give an example, a vocabulary size based speech recognition classification can be as follows:

- Small vocabulary - tens of words.
- Medium vocabulary - hundreds of words.
- Large vocabulary - thousands of words.
- Very-large vocabulary - tens of thousands of words.

Speech recognition systems are dealing with several source of variability that may exist in training and test speech which are difficult to model. These variabilities include:

- Intra-speaker variabilities,
- Inter-speaker variabilities,
- Presence of noise,
- Variable speaking rates,
- Presence of out-of-vocabulary words.

In case of speaker dependency and presence of noise, some adaptation techniques are used to increase the recognition accuracy. Adaptation issues will be given in chapter 7.

2.10.1 Small Vocabulary Speech Recognition

Small vocabulary speech recognition is considered as simplest speech recognition application. Generally this type of application are highly accurate, the error rates can be below 1%.

Small vocabulary speech recognizers are generally based on isolated word recognition or use a well defined deterministic grammar which can be modeled by a finite state automata.

This property make DTW and neural network methods ideal for small vocabulary speech recognition. Discrete HMM based speech recognition is also possible with large HMMs which model the entire words.

The decrease in accuracy of speech recognition in presence of noise is not as dramatic as in large vocabulary speech recognition systems if the acoustic realizations of words are not close to each other. Small vocabulary speech recognition systems are currently used on several real time applications including "name dialing" in mobile phones, small dialogs, call centers etc.

A potential domain of usage for small vocabulary speech recognition is embedded speech recognition which means use of speech recognition on low-performance hand-held devices or wearable computers. The need for applications that need less computing power and memory in such devices make small vocabulary speech recognizers ideal for this domain.

2.10.2 Connected Word Speech Recognition

Connected word recognition is based on a small vocabulary but unlike small vocabulary recognition the words are not isolated, two or more words can occur in one utterance with small silence intervals between them. The best example for this type of speech recognition is connected digit recognition like recognition of telephone numbers. In this example there is only 10 (or 11) words (digits) in the dictionary and the words can be combined in utterances of, for example, 7 to 13 words.

The performance of connected word recognition can be improved by using a language model which can be a finite state automata. This language model can restrict the occurrence of word sequence orders in a utterance. HMM word models or neural network based speech recognizers are ideal for this type of applications. The acoustic word models can be combined to obtain whole utterance. Language model can then be used to restrict the possible combinations.

In a dialog system with a small vocabulary, in connection with a well defined deterministic language model, connected word recognition can give good results. Since the number of competing word sequences will be low, the accuracy of this type of systems will be high.

Connected word recognition can be renamed as small vocabulary continuous speech recognition. In [30] a method for training of connected speech recognition is explained in detail. Unlike isolated word recognition, the transitions between words play an important role in the performance of connected speech recognition. In training phase, these transitions should be modeled well enough.

2.10.3 Large Vocabulary Speech Recognition

The speech recognition process explained in this chapter is covered completely in large vocabulary speech recognition. The other sub-classes include only certain parts of speech

recognition, for example only DTW based acoustic models for small vocabulary isolated word recognition task.

A large vocabulary speech recognizer is usually based on small sub-word units, generally phonemes, which are easier to combine for obtaining vocabulary words.

The basic architecture of a large vocabulary speech recognition system include the following blocks [31]:

1. Feature extraction,
2. Hypothesis search,
3. Lexicon,
4. Language model,
5. Acoustic model.

These blocks are explained in previous sections of this chapter. The main block which is responsible for recognition is the second block, hypothesis search. This block use the blocks, lexicon, language model and acoustic model, to recognize the speech content of the data coming from feature extraction block.

Lexicon includes the vocabulary words of speech recognition and their pronunciations. It is possible to have several phonetic representation of a word.

Language model includes probability of word occurrence in a context which can help hypothesis search block to limit the search only to valid word sequences.

Acoustic model provide a probability assignment to each feature vector which will be used by hypothesis search block to obtain a combined acoustic probability for a word hypotheses.

After obtaining all the inputs needed to make a decision, hypothesis search block makes a search to find the word sequence with highest likelihood. In large vocabulary speech recognition this search may include several word sequences which are in competition. It is possible to make an incremental search which consider only the sequences that will drive to high likelihood values.

Due to high complexity of the problem in large vocabulary speech recognition, the accuracies of the speech recognizers of this type are relatively low. The accuracy depends on several factors:

- Vocabulary size,
- Perplexity of language model,
- Presence of background noise,
- Spontaneity of speech,

- Low sampling rates, like in telephone speech,
- Insufficient amount of available training data
- Speaker dependency.

Researches on improved robustness of speech recognition are dealing with the problems occurred by these factors. Confidence measures, which will be detailed in chapter 5, try to give a measure of efficiency of the models used during hypothesis search. Poor modeling due to factors listed above, causes lower confidence levels on speech recognition outputs.

Adaptation is one of the technique used to obtain good recognition accuracies in case of mismatch between trained model and the test conditions. Both acoustic and language model can be adapted to new conditions in test environments.

2.10.4 Keyword Spotting

Keyword spotting is based on recognition of keywords in a acoustic word sequence. The interested part is only a portion of speech data. This method is a redefinition of speech recognition for particular application like "speech indexing" [32].

The base concept introduced in keyword spotting is garbage models. There are one or more garbage (sometimes called as filler) model to model the words that are not in the vocabulary of speech recognizer. An example of keyword spotting based speech recognizer can be as in Figure 2.14.

The hypothesis search include assigning probabilities to "garbage" models also. Since keyword spotting is focused only on some words in the spoken word sequence, the recognition performance is higher than large vocabulary continuous speech recognizers. The possible application areas could be speech indexing, spoken information retrieval, command and control systems etc.

The evaluation of keyword spotting systems, unlike other speech recognition methods, needs three statistic to be collected from recognition results:

- Correctly recognized words,
- False alarms,
- False rejections.

Different applications may give more importance to one of these statistics. In this case the decision threshold used by decoder can be updated to obtain the desired results. For example, if we make decisions harder, the false alarm rate will be decreased which can be a desired result when the retrieval of only related information is important for a spoken document retrieval system based on keyword spotting. It is possible to incorporate confidence measures on the evaluation of keyword spotting systems. Confidence annotation of recognized keywords may be helpful for a user of keyword spotting system.

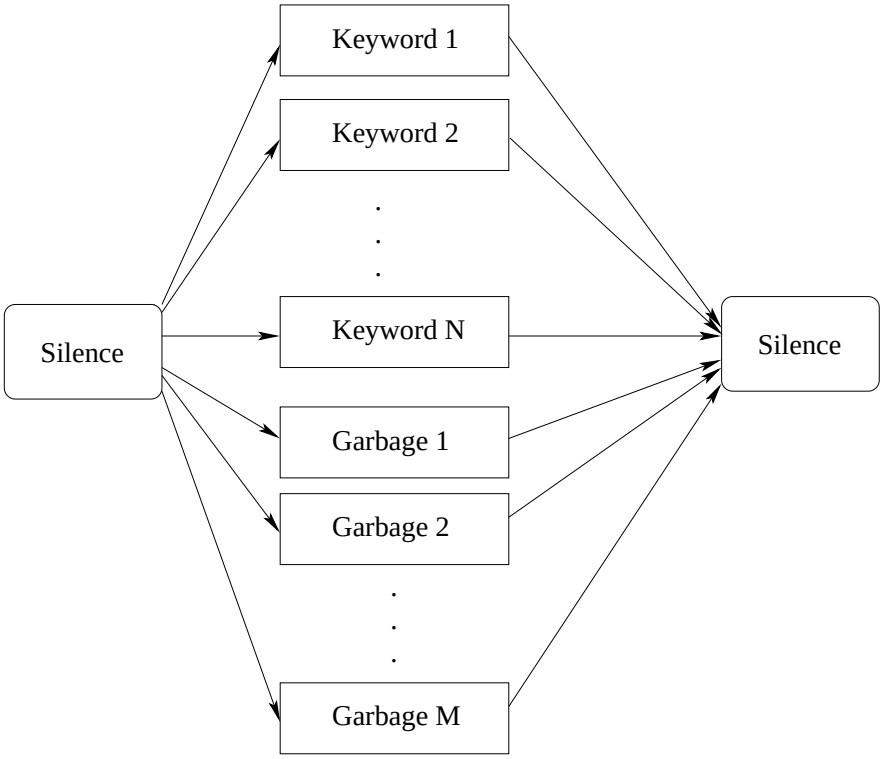

Figure 2.14: Keyword spotting based speech recognizer.

Chapter 3

Speaker Recognition

Speaker recognition is the process of automatically recognizing the identity of someone on the basis of informations obtained form the speech uttered by her/him. Speech contains characteristics of speaker which are unique. Everyone has a different voice. Voice is considered as one of biometric identification characteristics like fingerprints, DNA, iris, and face. All these biometric characteristics are used to find identity of someone.

Speech contain not only the message being spoken but also the information about the sound production system of the speaker. The statistical approach used in speech recognition can also be used in speaker recognition to obtain models of speakers instead of words.

Speech and speaker recognition fields are closely related. The feature extraction process used in speech recognition can efficiently be used in speaker recognition also. The principles of human speech recognition and human speaker recognition are considered to be same. Since the feature extraction methods explained in chapter 2 are based on human hearing principles, it is possible to use the features obtained for speaker recognition.

The advances in technology allowed us rapid processing of data which helps processing and classification of speech data in real time applications. Todays speaker recognition can easily outperform humans with training speech sequences of as low as 1 minute to obtain a speaker model. Real time speaker recognition application are increasingly being used in everyday life for improving security for example in building access systems, banking applications, etc. They are sometimes used in conjunction with other biometric identification techniques like fingerprint recognition or face recognition to improve the security.

Use of speech as the principle security control is still not preferred since it is possible to have some sources of variability in the speech which will reduce the recognition accuracy. When the application area is not high security demanding, the speaker recognition can be very helpful since the speech is the natural way of communication used by humans.

In this chapter we give first the information on biometric technology to better place speaker recognition technology, then we give a classification of speaker recognition followed by modeling techniques and model adaptation techniques.

3.1 Biometric Technology

Current biometric technology cover a variety of applications and it is difficult to make definition that cover all these applications. The most suitable definition of biometrics, as it is defined by "International Biometric Group", can be "the automated use of physiological or behavioral characteristics to determine or verify identity".

Physiological biometrics are based on direct measurements of parts of human body, like fingerprint, iris-scan, retina-scan, hand-scan, facial recognition and DNA sequence identification.

Behavioral biometrics are based on an action taken by a person. Behavioral biometrics, in turn, are based on measurements and data derived from an action. The measurements are indirect, that means only the effects of actions realized by body parts are measured and generally there are several body parts included in the actions. The main behavioral biometrics are voice, keystroke-scan and signature-scan. The function of brain in these actions is crucial. The characteristics obtained by these measures are closely related to coded actions in the brain. In behavioral biometrics time is an important metric that should be incorporated into measurements since every action has a beginning, a middle and a end.

An accurate and adequate data acquisition part is always important to obtain higher accuracies with biometric technology based identity determination.

3.2 Speaker Recognition

Speaker recognition is a behavioral biometric. Voice, the base for speech, is a function of vocal tract. In speech we can measure the articulation behavior of the speaker as well as the characteristics of vocal tract elements.

Speaker recognition techniques make it possible to use speaker's voice to verify their identity and control access to services. The possible areas of use are for example, voice dialing, banking over telephone line, access to confidential informations, voice mail, forensic applications, etc. The techniques are generally classified into two group:

- Speaker identification,
- Speaker verification.

Speaker identification is the process of determining the speaker's identity, speaker verification is used to verify the identity of a claimed speaker. Speaker verification can be seen as a special case of speaker identification in which the number of speaker to identify is only one. There is some small differences like use of a threshold for decision making in speaker verification but modeling techniques and similarity measures used for both systems are the same. The modeling techniques explained later in this chapter will be based on speaker verification which can be easily generalized to obtain a speaker identification system.

Speaker recognition systems are also classified with their dependency to the text content of speaker's voice:

- Text dependent systems,
- Text independent systems.

Text dependent speaker recognition is based on recognition of the text content, of the speech data obtained from the speaker. The text used for access is called a pass-word or a pass-phrase. Text dependent systems do not need modeling of the speaker, it is possible to use a speech recognition system as this type of system. It is possible to train a simple phoneme based speech recognizer for text dependent speaker recognition. Since the vocabulary size will be limited we can expect high performance systems. The problem with text dependent system is that if the password is stolen then it is possible for impostors to have access rights. Text prompted speaker recognition tries to alleviate this problem by changing the password at each access of a speaker [37].

Text independent systems are based on speaker dependent characteristics that may exist in the voice. Speaker modeling is important for this type of speaker recognition systems. The system obtained by combination of text dependent and text independent methods generally give better performance. The most secure systems are obtained by using text independent speaker models in conjunction with prompted text recognition.

3.2.1 Speaker Identification

In speaker identification, the task is to compare a test utterance with several models stored in the system and to find the best matching model which represent the identity of true speaker. The stored models can contain some "garbage models" which can be used to reject some utterances if the test utterance match one of these models.

Figure 3.1 shows an example of a speaker identification system. The speaker models and garbage models are obtained on some classified training data. The modeling techniques used in training phase will be detailed later in this chapter.

Speaker identification systems have a limited number of speakers known to system whose are represented by their models trained on some previously collected training data for each speaker and some garbage models which are trained on some data collected from outside world.

The garbage models can be trained by the data from several known speakers. In this case the garbage model called also "world model" is a generalized speaker model. If this model is matched by the identification system, that means the system could not match correctly the input speech to any known speaker, so the speaker is "unidentified".

The accuracy of speaker identification system can be measured simply by counting the number of incorrect assignments. Some application may interest in the rate of identifying one

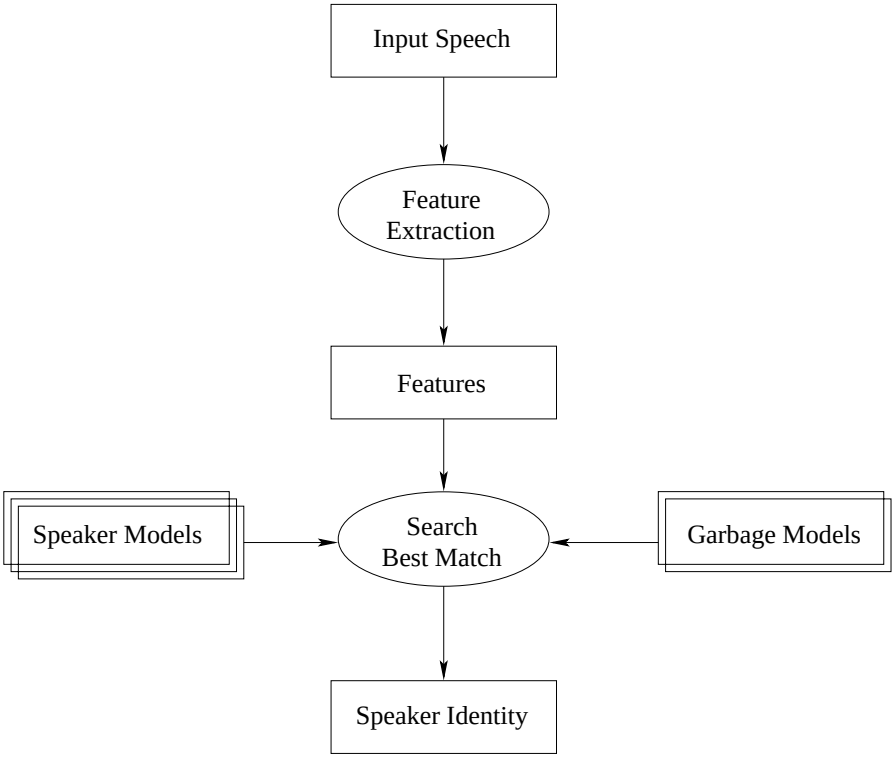

Figure 3.1: Speaker identification system

speaker as another one or the rate of identifying an impostor as one of known speaker, in this case the performance of speaker identification system can be summarized by a graphic called Receiver Operation Characteristic (ROC). This graphic, called "ROC curve", is obtained by plotting *false alarm rates* on the horizontal axis and *correct detections rates* on the vertical axis.

A variant of ROC curve is, introduced by [33], Detection Error Tradeoff (DET) curve. In DET curve both axes represent error rates: False alarm rates and miss rates. Equal error rates (EER), which are the error rates where false acceptation rate and miss rates are equal, estimated from DET curves can give a good estimation of speaker identification performance of the system.

The performance of speaker identification systems depends on population size and the quality of speech data available for the task. More available training data results in better modeling of speakers when the models are statistical. Speaker modeling issues will be detailed later in this chapter. Some results for speaker identification using Gaussian mixture models can be seen on [34].

Capture of inter-speaker variabilities is very important for improved performance of speaker identification systems. Cohort normalization is one of the methods used for performance improvement. It is based on creating similarity groups between known speakers and increasing the discrimination abilities of the speaker models inside these groups. For a

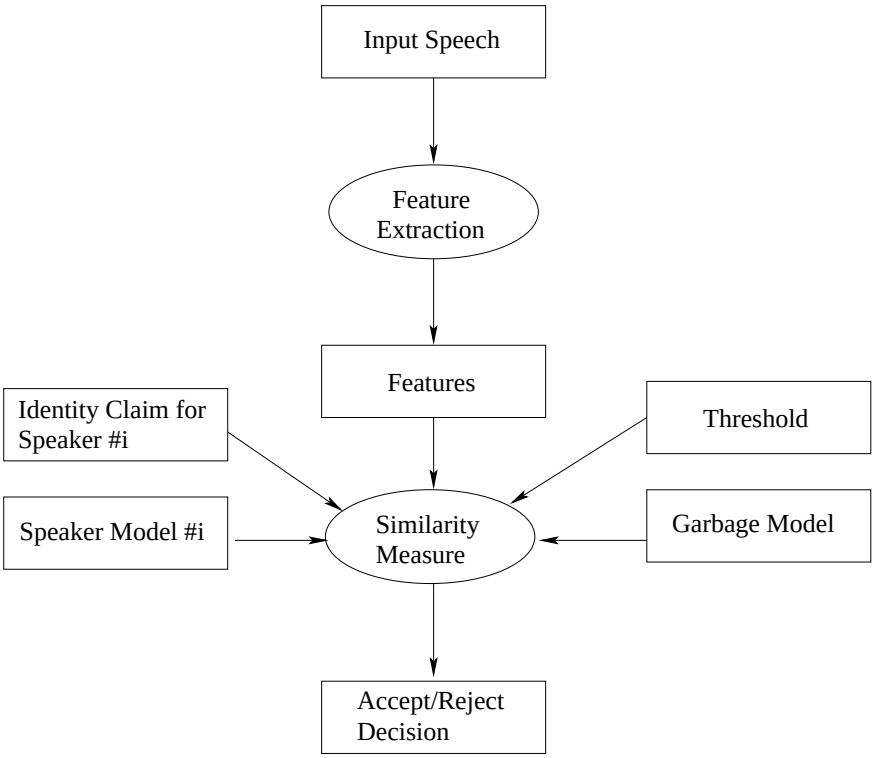

Figure 3.2: Speaker verification system

specific speaker, a cohort model can be created by finding other speaker models which are close or similar to the model of that speaker. A good cohort selection and normalization method could improve significantly the EER [35], [36].

3.2.2 Speaker Verification

Verifying the identity of a speaker by his voice is called speaker verification. This technique is used generally for access control purposes. The speaker claims his identity and speaks to the system, the system compare the speech data with the model of claimed speaker and give access right or reject him/her according to a certain threshold of acceptance. A typical speaker verification system is shown on Figure 3.2. The difference between a speaker identification and speaker verification system is mainly the search procedure which does not exists on the later. In speaker verification, the task is easier since the speaker is claiming his identity and the system knows which model will be used for comparison. In the test phase, the only thing to do is measuring the similarity between the model of speaker and the speech data, then comparing it to a threshold.

Training phase of speaker verification include, unlike speaker identification, threshold determination which consists in fixing a threshold value for each speaker, or a general threshold, which will be used in test phase to take a decision. As the identity of speaker is “known”, the

output is "access" or "reject" decision. A garbage model can still be used as a double check in conjunction with the threshold. The garbage model which is also called "world model" can be replaced with cohort model of each speaker. Use of cohort models, as explained for speaker identification, make speaker verification more sensible to the small discriminations that may exists between speakers with similar voices [36].

Speaker verification can be text dependent or text independent. In text dependent systems, as will be seen later in this chapter, the phonetic content of speech is also used to verify the identity of speaker.

3.3 Speaker Modeling

The first phase of a speaker recognition system is the training phase which include creating and training models for each known speaker. World model, garbage models and cohort models are also created in training phase.

Speaker modeling techniques can be classified into two groups:

- Template models,
- Stochastic models.

Template models are based on creating some template feature vectors for each speaker. They are text dependent. The templates are created by normalizing the feature vector sequence of the pass-word which is recorded on several session. Dynamic Time Warping (DTW) technique is used for similarity measures in template model based speaker recognition.

In **Stochastic models**, speech production is assumed as a parametric random process which can be modeled by parameter estimation of the underlying stochastic process. As it was in speech recognition, stochastic modeling techniques give better results.

In speaker recognition context, if the speaker models are text dependent which means the text content of the speaker voice is also represented in the models, HMM modeling is used. If the speaker recognition is text independent, single state HMMs called Gaussian mixture models (GMM) are used to create speaker models. This technique, as will be explained later, provide good accuracies which are comparable with text dependent systems with increased security [40].

The feature extraction phase of training will not be detailed here since it is the same as it was for speech recognition. MFCC based feature extraction can also be used in speaker recognition applications.

3.3.1 Dynamic Time Warping Based Speaker Models

Dynamic time warping (DTW) is used for template matching problems. The use of DTW for speech recognition is explained on chapter 2. For speaker recognition there is no

significant differences in the usage of DTW technique. Unlike speech recognition which is based on templates of words, in speaker recognition there are templates for each speaker.

Training of a DTW based speaker recognition system can include following steps:

- Feature extraction for each utterance,
- Reference template construction,
- Determination of decision threshold.

In training phase, for each speaker 3-5 utterances of same pass-word is obtained, after feature extraction of each utterance, some normalizations are applied to have a better reference template of the utterance. A decision threshold is determined according to the distance between reference template and all utterances. The distances with reference templates of other speakers can also be considered to model inter-speaker variabilities as explained in [38].

The reference template $\overline{X}$ is the mean of training utterances and is defined as:

$$\overline{X} = \overline{x}_1, ..., \overline{x}_N \tag{3.1}$$

where N is the length of reference template represented with a feature vector sequence. In test phase this reference template is compared with the test utterance to obtain a similarity score. If the test utterance is:

$$X = x_1, ..., x_M \tag{3.2}$$

where M is the length of the test utterance, the similarity score can be obtained by:

$$z = \sum_{i=1}^{M} d(x_i, \overline{x}_{j(i)}) \tag{3.3}$$

where z is the similarity score, $d(\ ,\ )$ is the distance function used in DTW algorithm and $j(i)$ is the template index given by DTW algorithm [39]. DTW algorithm is summarized in equation (2.9).

DTW based speaker modeling can be improved by using VQ codebooks. Use of VQ codebooks increase the generalization capability of DTW based modeling but need much more training data for VQ codebook training.

3.3.2 Gaussian Mixture Models

Gaussian mixture modeling (GMM) is a statistical method for creating models which can be used in statistical pattern matching problems. Since speech can also be defined as a parametric random process, use of statistical pattern matching techniques is suitable for speaker recognition as they are being used in speech recognition. The feature vectors obtained after feature extraction provide informations about the text content of speech signal as well as some speaker related characteristics such as vocal tract shape or articulation.

In statistical speaker recognition, the goal is to create a statistical model for each speaker which is used then to determine if a particular speaker has pronounced the test speech signal. The likelihood probability $P(S_i|O)$ determines the probability that the speaker model S_i created the observation O. This probability cannot be evaluated directly, we need to use Bayes' theorem to rewrite this probability as:

$$P(S_i|O) = \frac{P(O|S_i)\ P(S_i)}{P(O)} \tag{3.4}$$

where $P(O|S_i)$ is the probability that observation O is created by speaker model S_i which can now be evaluated easily, $P(S_i)$ is the probability that speaker i will be recognized and $P(O)$ is the probability of observing the observation sequence O.

The probability $P(S_i)$ can be considered as uniform for all speaker since the system does not have any prior information about "which speaker is testing the system" at a given time, and it is ignored. $P(O)$ can be computed by summing the probability that observation O is generated by any of the speaker model in the system, $P(O|S_j)$. The equation (3.4) can be rewritten as:

$$P(S_i|O) = \frac{P(O|S_i)}{\sum_{j \in I} P(O|S_j)} \tag{3.5}$$

where I is the set of all speakers.

In equation (3.5) the computation of the likelihoods for all speakers is not feasible when the speaker set is large. Instead of computing all of the likelihoods, some smaller speaker groups are defined by grouping similar speakers in a group. This technique is called cohort modeling. When cohort modeling is used, the number of likelihood computation is reduced to a small subset of I for each recognition process. It is possible to create cohort models for each speaker and store them in the system if the memory constraints are not important. In this case the equation (3.5) can be modified as:

$$P(S_i|O) = \frac{P(O|S_i)}{P(O|S_{C_i})} \tag{3.6}$$

where S_{C_i} is the cohort model for the speaker S_i. The use of a world model which include all the speakers instead of only the speakers in a small group is also used for speaker recognition. When the world model is used, the probability of observing O can simply defined as $P(S|O)$ where S is the world model, the equation (3.5) can then be rewritten as:

$$P(S_i|O) = \frac{P(O|S_i)}{P(O|S)} \tag{3.7}$$

World modeling is more efficient then using cohort models since the system need less memory to keep the models and the search is faster.

The feature vectors are assumed to be generated according to a multidimensional Gaussian probability density function (pdf) with a state dependent mean and a covariance. The

Gaussian pdf for a D-dimensional feature vector x is defined as:

$$b_i(x) = \frac{1}{\sqrt{(2\pi)^D \ |\Sigma_i|}} \ e^{-\frac{1}{2}(x-\mu_i)^T(\Sigma_i)^{-1}(x-\mu_i)} \tag{3.8}$$

where μ_i and Σ_i are the mean and covariance matrix of the i^{th} component of the Gaussian mixture, $(\ )^T$ means matrix transpose.

The $b_i(x)$ in equation (3.8) is the probability density of Gaussian component i. The likelihood probability of feature vector x given speaker model S_j, $P(x|S_j)$, can be obtained by the weighed sum of M Gaussian densities as:

$$P(x|S_j) = \sum_{i=1}^{M} p_i b_i(x) \tag{3.9}$$

where p_i is the the mixture weight of component i. Each component density is a D-variate Gaussian function.

The parameters of Gaussian mixture speaker model S_j are defined as:

$S_j = \{p_i, \mu_i, \Sigma_i\}, i = 1, ..., M.$

The sum of mixture weights satisfies the criteria,

$$\sum_{i=1}^{M} p_i = 1$$

The model parameters are obtained by Expectation Maximization (EM) algorithm which is based on Maximum Likelihood (ML) estimation during training phase [41].

At the end of training phase we obtain speaker models, S_j, which will then be used by a maximum likelihood classifier for speaker recognition purpose. The classification can be applied by maximizing the equation (3.7) when a world model is used or the equation (3.6) when the cohort models are used for normalization of likelihoods.

It is possible to work in *log* domain since the problem is a maximization problem. We can rewrite the equation (3.7) as:

$$\Lambda_i(O) = \log P(O|S_i) - \log P(O|S) \tag{3.10}$$

where $\Lambda_i(O)$ is the log likelihood ratio and is a replacement for $\log P(S_i|O)$.

In speaker verification task, $\Lambda_i(O)$ is compared to a threshold θ which is determined during training. If $\Lambda_i(O) > \theta$ then the speaker is verified, otherwise he/she is rejected. In speaker identification the task is searching the most likely speaker model that have created the observation O. It can be formulated as:

$$\hat{n} = arg \max_{1 \leq n \leq N} \Lambda_n(O), \tag{3.11}$$

where $\hat{n}$ is the identity of the speaker, N is the number of known speakers.

Speaker recognition using GMM techniques, as shown in [42] and [34], give good enough results that can be used in real life applications like access control authentication. It is concluded from [42] that, performance of GMMs does not depends on initialization, they can learn speaker specific characteristic from training data. The GMMs are performant in the presence of corrupted speech.

The main limitation with GMMs may be the performance reduction when the system is used in mismatched conditions. In this case some adaptation techniques are applied to improve the robustness of speaker models. Adaptation issues will be given later in this chapter.

The GMM based speaker recognition is generally text independent and can be combined with a speech recognition method to obtain text dependent speaker recognition systems. When the text content is important the HMM techniques which which are more powerful techniques capable of modeling text content together with speaker specific characteristics.

3.3.3 HMM Based Speaker Models

HMM technique is used for speaker recognition when the system is text dependent. The pass-words or pass-phrases of each speaker are modeled by HMMs. The modeling task is similar to use of HMM in speech recognition as it was explained in chapter 2. Since the vocabulary is limited to the number of speakers, there is no need to use of phoneme based HMMs which are useful in large vocabulary speech recognition.

HMM can be seen as a generalized version of GMM which is simply a one state HMM. In HMM the forward-backward algorithm [5] is used for parameter estimation in conjunction with EM algorithm since the mixture weights depend on previous states or previous time frames. Speaker models obtained by HMM techniques take into account the a priori time dependencies between different time frames.

The initial HMM model during training can be obtained by creating one state for every phoneme in the pass-phrase. The pass-phrase speech data is needed to be phonetically labeled for training.

In verification phase, the HMM score which is the likelihood probability that the model of the claimed speaker generated the test speech data is computed by applying Viterbi decoding algorithm. Viterbi algorithm finds the best state path to obtain the highest possible likelihood score for an HMM.

An HMM based speaker verification, can be obtained simply by training speaker dependent HMMs for pass-phrases assigned to speakers. Figure 3.3 shows an HMM based speaker verification system with training and verification phases. In training phase, for each speaker a pass-phrase HMM is constructed and it is trained by several repetition of pass-phrase utterance obtained from the speaker. These utterances are then re-scored by the speaker HMM to obtain a threshold acceptation score for the speaker. In verification phase, the phrase

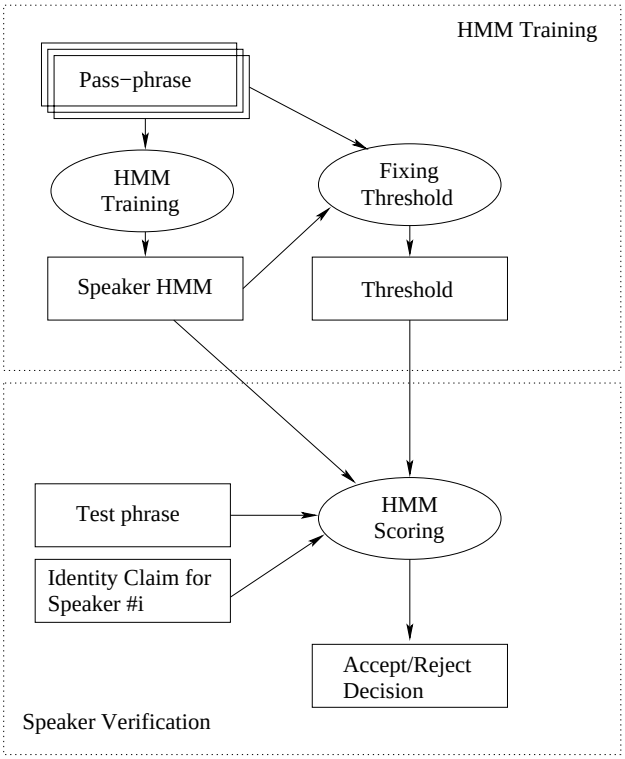

Figure 3.3: An HMM based speaker verification system

obtained from the speaker is scored by the model of claimed speaker, if the score is higher than the speaker threshold, access decision is taken.

The HMM technique can be combined with neural networks to obtain a hybrid speaker recognition system. The use of HMM/ANN hybrid systems is same as it is used for speech recognition. An example of HMM/MLP hybrid speaker verification is implemented in [43]. The speaker verification with HMM/MLP hybrid system is summarized as:

1. Each speaker pronounces 2 or 3 times his/her password.
2. A phonetic transcription of each utterance is obtained manually or by using a speaker independent speech recognition system.
3. The speaker HMMs are initialized from phonetic transcriptions.
4. An MLP for each speaker is trained, thresholds are determined.
5. MLPs are used for estimating HMM probabilities in Viterbi algorithm to obtain the speaker verification score.

The speaker MLPs can also be obtained by adapting a speaker independent MLP which is used for speech recognition purposes.

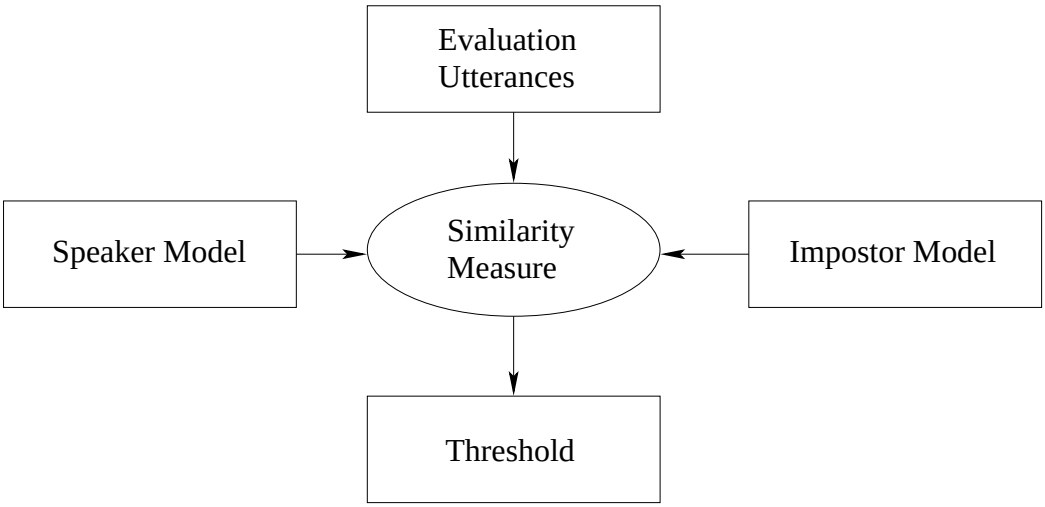

Figure 3.4: Threshold determination using impostor models.

3.4 Thresholds and Access/Reject Decisions

Threshold usage is different in speaker identification and speaker verification tasks. In speaker verification the similarity measure between observation sequence and claimed speaker model is compared with a threshold. This similarity measure is usually the likelihood ratio score obtained by normalizing the probability that speaker model created the verification utterance by the probability that a world/cohort model created the verification utterance. Generally the logarithm of this likelihood score is used. This score can be written as:

$$\Lambda(O) = \log P(O|S) - \log P(O|\overline{S}) \tag{3.12}$$

where $\Lambda_i(O)$ is the the log likelihood score used as a similarity measure for speaker recognition systems. S is the speaker model and $\overline{S}$ is the world/cohort model. In some speaker verification systems $\overline{S}$ is trained by some known impostor data as shown on Figure 3.4.

Accept/Reject decisions are based on speaker thresholds or general thresholds. If likelihood score computed for a speaker model is above a threshold, the accept decision is taken, otherwise, the reject decision is taken. In speaker identification use of threshold is not necessary since the task is searching for best matching model. If garbage models are not used, it is possible to reject some speaker whose maximum score is smaller than a certain threshold.

In speaker verification, threshold plays an important role for decision making process:

- A high threshold makes it difficult for impostors to be accepted by the system while making possible the rejection of correct speakers.
- A low threshold allows system to accept more correct speaker while allowing accepting some impostors also.

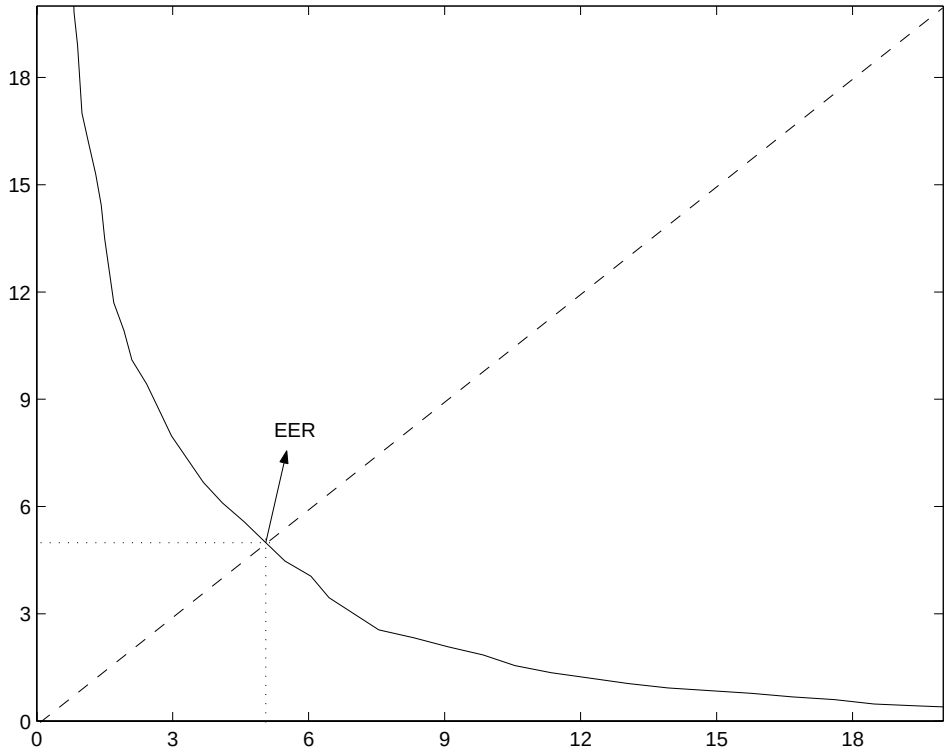

Figure 3.5: DET curve. Equal error rate (EER) is the intersection of the curve with the line $x = y$.

A good threshold should be aware of the distribution of correct speaker scores and impostor scores.

The effectiveness of a speaker verification systems can be evaluated by ROC curves. A ROC curve, as explained before, is obtained by plotting false acceptance rates versus correct acceptance rates [44]. The point on the curve are obtained by varying decision threshold.

DET curve is a variant of ROC curve on which axes are false rejection (miss) error rates and false acceptance error rates. An example DET [33] curve is shown in Figure 3.5. DET curves make it easy the extraction of a useful statistic called Equal Error Rate (EER). The variations on the DET curve can be analyzed to evaluate the correctness of EER. EER is accepted as overall performance measure of a speaker recognition system [44].

The EER based threshold determination can be summarized as:

1. Train speaker models,
2. Plot DET curve for each speaker model by using an evaluation data set or the training set,
3. On the DET curve, take the EER point which is the intersection between the line $x = y$ and the DET curve,
4. Take the threshold used to obtain EER as the threshold of the system.

Threshold can be selected as greater than EER or smaller than EER threshold according to desired level of security. Thresholds can be modified without changing the general structure of the speaker verification system.

3.5 Speaker Model Adaptation

Model adaptation is the action of adapting speaker models, speech recognition models, or other models used in speech/speaker recognition tasks to recognition conditions or to specific speakers for the purpose of obtaining better recognition results. In speaker recognition two type of models are used, speaker models and world/cohort models. Model adaptation in speaker recognition consists in adapting these models to the conditions of application environment.

Speaker adaptation techniques are used to compensate acoustic mismatch between training and recognition conditions. For example, if the training data is obtained in a laboratory environment or in a silent room condition, when the system is used for recognition the performance may be lower than expected level because of the presence of noise in the acoustic data used in recognition phase. Recognition is usually realized in "real" environments which are open to various sources of variability. To alleviate the reduced performance problems in "real" conditions, the speaker models are adapted to these conditions by doing some more training in recognition environments.

Speaker adaptation techniques can be classified into three groups [45]:

- Feature transformation,
- Score normalization,
- Model transformation/adaptation.

Feature transformation technique is used for "cleaning" the noise from acoustic data to obtain feature vectors that can be used for speaker models trained on clean speech. One of feature transformation technique, cepstral mean subtraction, is applied by subtracting the average feature vector from each feature vector in order to remove effects of environmental stationary signals from the speech signal.

Score normalization can be applied by normalizing speaker model scores for the verification utterance with world/cohort model scores. This technique gives likelihood ratio scores as explained in previous sections.

Model transformation/adaptation is based on updating the parameters of speaker models in order to obtain a speaker model which covers the new (adaptation) data. In this chapter we will focus on model adaptation for GMM based speaker recognition.

The main model adaptation techniques used for adaptation of GMM based speaker recognition are:

- Maximum A Posteriori (MAP) based adaptation,
- Maximum Likelihood Linear Regression (MLLR) based adaptation,
- Eigenvoices based adaptation. [47].

In MAP adaptation, the Gaussian means of initial model are updated as a function of adaptation data.

In MLLR adaptation, the adaptation data is used to obtain a linear regression transformation matrix for the Gaussian means. This matrix is then used to update the Gaussian means of the initial speaker model.

In Eigenvoices based adaptation is applied by estimating eigenvectors for feature vectors of adaptation data from Principal Component Analysis (PCA) technique [46]. These eigenvectors are called as Eigenvoices and are used to update Gaussian means of speaker models. Use of Eigenvoices for speaker adaptation, as claimed in [47], gives comparable results to MAP and MLLR adaptation techniques.

MAP and MLLR adaptation techniques will be detailed in chapter 7.

Speaker adaptation techniques can be used to add speakers to the system by creating new speaker models. For this purpose the world model can be used as a base. Training data for the new speaker can be used for determination of adaptation weights, the world model is then transformed by this weights to obtain the model of new speaker.

Model adaptation, if carefully applied, can be a powerful method for maintaining and improving the performance of models used for speaker verification. The adaptation must be carefully applied since adaptation with the data of an impostor could be harmful to the system.

3.6 Use of Confidence Measures in Model Adaptation

The confidence measure technique can be used to evaluate the reliability of the recognition results. From this definition it is clear that if we can delete low confidence data from adaptation data, the efficiency of adaptation process will be improved.

Confidence measures are usually based on a posteriori probabilities or likelihood scores generated by a speaker model applied to some observation data. Average posterior probabilities and likelihood ratios are most commonly used confidence measures. In likelihood ratio method, the observation data can be rescored by an impostor model which is created during training phase from some predefined impostor data set.

In unsupervised speaker adaptation, confidence measures can be used as a adaptation data selector. When the confidence on the accept decision is high enough, the adaptation can be applied, else the model remains unchanged. Confidence measures can be useful in supervised speaker adaption to verify if the quality of newly collected data is sufficient.

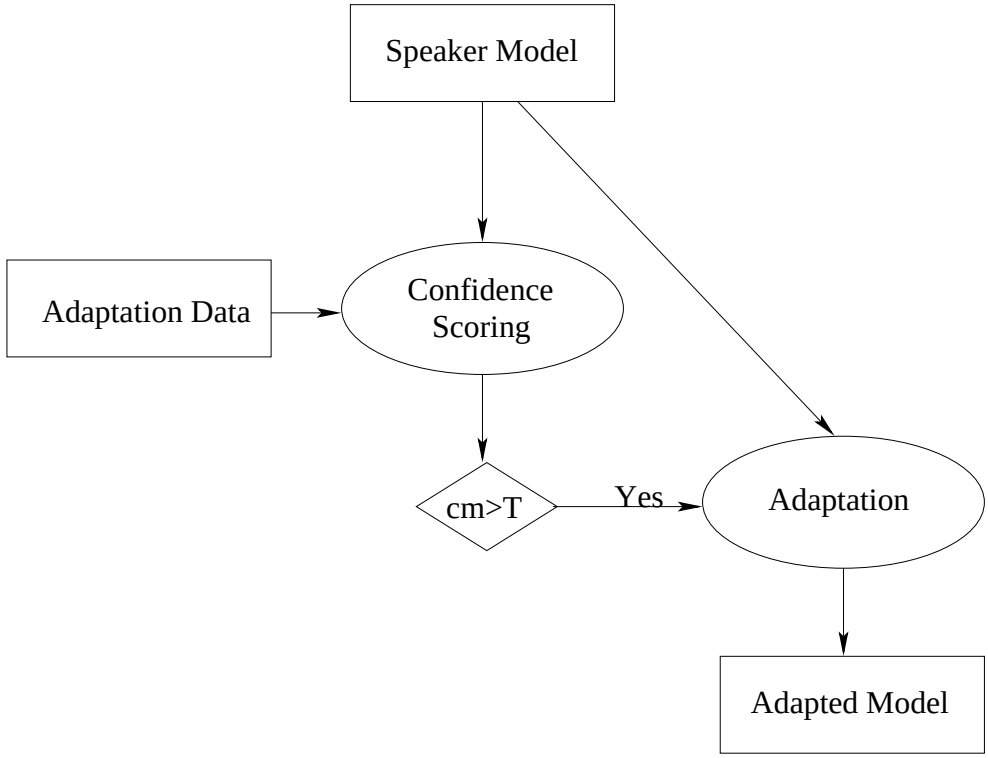

Figure 3.6: Speaker model adaptation using confidence measures. T is a threshold and cm is the confidence score for the adaptation data.

Use of confidence measure in unsupervised speaker model adaptation can be as in Figure 3.6.

Confidence measures can be useful in adaptation tasks, since they don't allow usage of adaptation data that is very different than the training data used to obtain the initial model. This restriction make the adaptation more secure.

Confidence measure issues will be revisited in chapters 6, 7 and 8 with proposed techniques of this thesis and some results.

Chapter 4

Turkish Large Vocabulary Continuous Speech Recognition (LVCSR)

Turkish is an agglutinative language. An agglutinative language is defined as a language in which words are made up of a linear sequence of distinct morpheme and each component of meaning is represented by its own morpheme. This means a complete sentence can appear as one word. In agglutinative languages it is possible to produce a very high number of words from same word with affixes [48].

The Turkish language is classified into the Ural-Altaic language class which has the following features:

- Vowel harmony. There is a vowel harmony between the first syllable and the following syllables in the a word. The suffixes have different vowels according to the last vowel in the word. If the last vowel in the word is a back vowel then the suffixes are formed by back vowels, if it is a front vowel then the suffixes are formed by front vowels. For example in Turkish, the plural of "banka" (the bank) is "bankalar" where the plural of "kedi" (the cat) is "kediler". The plural suffix, "-ler" changes the form according to the last vowel in the word.

- Absence of gender. There is no gender for objects like in English but there is no gender for the persons neither.

- Agglutination. The most distinctive feature of this class of language is the agglutination. This feature is important in continuous speech recognition, especially for language modeling, since the unlimited word creation is the result of this feature.

- The adjectives precede nouns.

- Free constituent order. The order of words can be changed according to the prosody.

All these features of Ural-Altaic languages make their Natural Language Processing (NLP) different than Indo-European languages like French, English, German etc. NLP is the last step of an ideal speech recognition system which helps understanding of speech.

Language modeling of these languages must be different then the language modeling techniques explained in chapter 2. The free word order and the high number of words in these languages make it difficult to obtain a word-order based efficient language model. This issue will be revisited later in this chapter and a new language modeling will be proposed for agglutinative languages.

4.1 The Turkish Language

Turkish is a logical language of the Ural-Altaic family with few exceptional rules and no genders, but its agglutinative structure is different from Indo-European languages for example German language which is considered also as agglutinative.

A Turkish word starts with a short root (such as git-, 'go'), then one or more suffixes are added to modify the root (gitti, 's/he went'). English uses only a few suffixes, such as -'s for possessive, -s or -es for plural, but Turkish has dozens of suffixes. You can make whole sentences in Turkish out of one little word root and a lot of suffixes.

In Turkish, similarities between morphemes has a negative effect in speech recognition. The morphemes convey the morphological informations like tense, case, agreement, etc., and they are concatenated to a free morpheme or a root word. Since the morphemes are concatenated to each others in order to create new words that convey morphological information, the number of different words in the language is unlimited. According to [48] the number of distinct words in a corpus of 10 million words is greater than 400,000, which is a problem for classical n-gram language modeling, due to data-sparseness problem.

The unlimited number of words may cause a high Out-of-Vocabulary (OOV) word rate for LVCSR tasks. It is highly probable that a large number of words may not appear in the vocabulary of the speech recognizer even if the vocabulary contains more than 100,000 words. For example a word like:

AMERIKALILAŞTIRAMAYABİLECEKLERİMİZDENMİŞSİNİZCESİNE

which means "as if you were those whom we might consider not converting into an American" is difficult to observe in a corpus used for language model training. The suffixes does not have only inflectional but also derivational effects on the root words. This feature yields to creation of such long words which are actually complete sentences. For the word above the inflections and derivations can be:

AMERIKA	: America
AMERIKALI	: The person from America

AMERIKALILAŞMAK : Being converted into an American. (-MAK for infinitive form of the verb)
AMERIKALILAŞTIRMAK : Converting into an American.
...

Another feature that may reduce the recognition accuracy is the existence of confusable suffixes like in following example:
kedi + n (your cat)
kedi + m (my cat)
kedi + me (to my cat)
kedi + ne (to your cat)
kedi + n + de (from your cat)
kedi + m + de (from my cat)

This example shows that in each pair the only distinctive phonemes are **-n** and **-m** which are both nasal and are confused in recognition.

It is possible also to observe use of same endings, group of suffixes, with different root words. When the root words are small, there will be a confusion between word models. For example:
ev + lerimizdekiler (the persons who are in our houses)
iş + lerimizdekiler (the persons who are in our work)
in + lerimizdekiler (the persons who are in our caves)
ön+ lerimizdekiler (the persons who are in our front)

The small roots in this example have a high probability of confusion with other words.

Although there are various disadvantage, the word creation and morphological structures in the Turkish language are rule-based and can be modeled by the finite state approach [49]. There is hardly any exceptions to some fundamental principles. These principles are:

1. Stems and affixes are unambiguous in their morphophonemic form,
2. One affix represents only one grammatical category.

The suffixes can be grouped into two general group:

- Inflectional suffixes. The suffixes that does not change the meaning of the word. These suffixes provides syntactic context of the word.
- Derivational suffixes. The suffixes that create a new word when concatenated to a word. The meaning of newly created word usually is related to the original word.

Derivational suffixes always comes before inflectional suffixes.

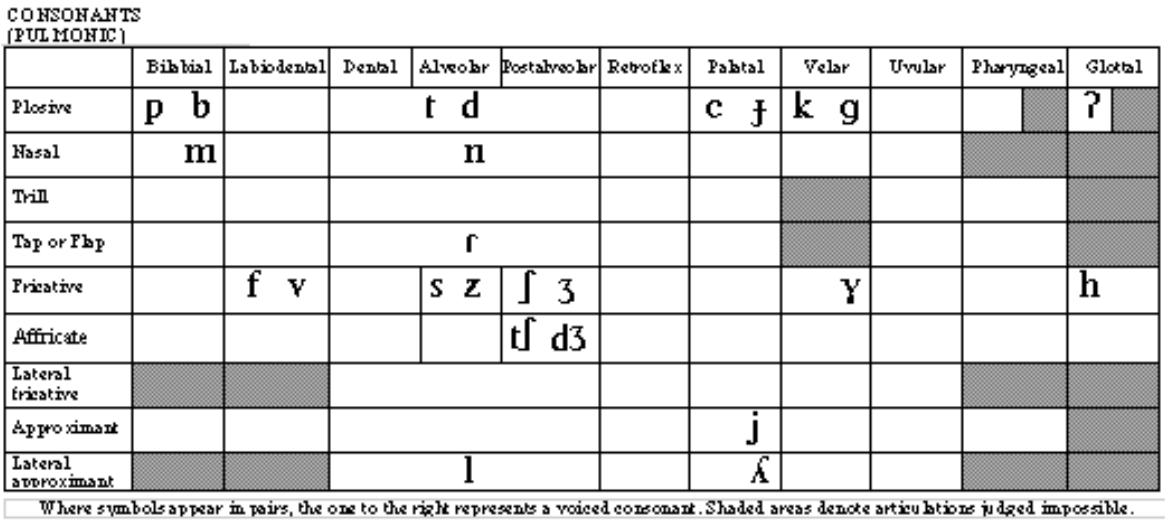

Figure 4.1: IPA chart for Turkish consonants

The syntactic informations are encoded with morphemes and are included in the word. For example, the sentence:
"I have to go to hospital"
is written as:
"Hastaneye gitmeliyim"
in Turkish, by simply adding morphemes to the root words "hastane" (hospital) and "git" (to go).

Turkish has a phonemic orthography. The vowels and consonants in Turkish are shown in Table 4.1 with their phonemic representation used in this thesis, Speech Assessment Methods Phonetic Alphabet (SAMPA) symbols and International Phonetic Alphabet (IPA) definitions [51], [50] [1].

IPA representations of phonemes are also shown on Figure 4.1 and on Figure 4.2 [50]. Note that some consonants on Figure 4.1 are not represented in Table 4.1 since they are from some dialects of the Turkish. The symbols which can appear in Turkish dialects are, unvoiced plosive palatal (k), voiced plosive palatal (g), plosive glottal(k) and lateral aproximant palatal (y).

4.2 Morphology of the Turkish Language

In linguistics, morphology is defined as the structure of word forms. Words are the combinations of discrete meaningful units called "morphemes". Morphemes can be "free" or "bound". Free morphemes can constitute a word on their own, while bound morphemes

[1] http://classweb.gmu.edu/accent/nl-ipa/turkishipa.html

Table 4.1: Turkish vowels and consonants with their phonemic representations used in this thesis, their SAMPA symbol and IPA classification.

orthographic	Phoneme	SAMPA	IPA name
vowels			
a	a	a	open-back
e	e	e	close-mid-front
ı	@	1	close-central
i	i	i	close-front
o	o	o	close-mid-back-rounded
ö	@h	9	close-mid-front-rounded
u	u	u	close-back-rounded
ü	y	y	close-front-rounded
consonants			
b	b	b	plosive bilabial-voiced
c	dz	dZ	affricate post-alveolar-unvoiced
ç	ts	tS	affricate post-alveolar-voiced
d	d	d	plosive alveolar-voiced
f	f	f	fricative labiodental-unvoiced
g	g	g	plosive velar-voiced
ğ	: (gh)	: (G)	ignored (fricative velar)
h	h	h	fricative glottal
j	zh	Z	fricative post-alveolar-voiced
k	k	k	plosive velar-front
l	l	l	lateral aproximant alveolar
m	m	m	nasal bilabial
n	n	n	nasal alveolar
p	p	p	plosive bilabial-unvoiced
r	r	r	trill (tap or flap) alveolar
s	s	s	fricative alveolar-unvoiced
ş	sh	S	fricative post-alveolar-unvoiced
t	t	t	plosive alveolar-unvoiced
v	v	v	fricative labiodental-voiced
y	j	j	aproximant palatal
z	z	z	fricative alveolar-voiced

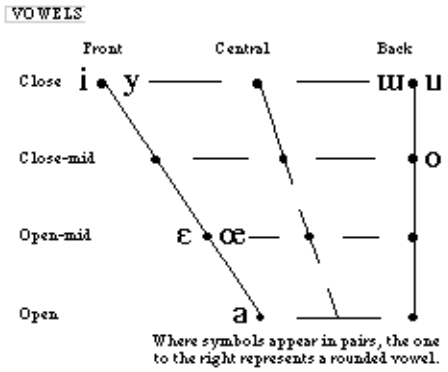

Figure 4.2: IPA chart for Turkish vowels

must appear with one or more morphemes to form a word. Words often consist of a free morpheme and one or more bound morphemes.

For example, the word "unbelievable" has three morphemes "un-", a bound morpheme, "-believe-", a free morpheme, and "-able". "un-" is also a prefix, "-able" is a suffix. Both are affixes.

The free morphemes are the bases of words, it is possible to attach bound morphemes to them for creating new words. Free morphemes are called also as root morphemes. Bound morphemes can be classified into two groups:

- Derivational morphemes,
- Inflectional morphemes.

Derivational morphemes are used to create new words from words and the meaning of newly created word is different from the base word.

Inflectional morphemes are used to change the word form without changing the meaning of the word. For example, the morphemes used for conjugation, like "-ed", are inflectional morphemes.

As stated before, Turkish is an agglutinative language. The word structures are formed by productive affixations of derivational and inflectional suffixes to root words. The extensive use of suffixes causes morphological parsing of words to be rather complicated, and results in ambiguous lexical interpretations in many cases. For example[2]:

-*çocukları*
a. child+PLU+3SG-POSS his children
b. child+3PL-POSS their child

[2]The morphological tags used here are; PLU: plural form, 3SG: third person singular verbal agreement, POSS: possessive, 3PL: third person plural verbal agreement, ACC: accusative case, GEN: genitive case, PAST: past tense

c. child+PLU+3PL-POSS their children
d. child+PLU+ACC children (accusative)

Such ambiguity can sometimes be resolved at phrase and sentence levels by the help of agreement requirements but this is not always possible:

Onların çocukları geldiler
(He/she)+PLU+GEN child+PLU+3PL-POSS come+PAST+3PL Their children came

In the example above, the correct parsing for the word *"çocukları"*, is (c) since the agreement is required between the subject and the verb. But in the example:

Çocukları geldiler.
child+PLU+3SG-POSS come+PAST+3PL His children came.
child+PLU+3PL-POSS come+PAST+3PL Their children came.

The ambiguity exists in this example since the correct parsing may be either (a) or (c) while the subject can be either "onun" (his) or "onların" (their).

Morphological ambiguity also results in a single lexical item having different parts-of-speech which may sometimes be resolved via statistical or constraint-based methods though this is not always possible and have to be dealt with at the syntactic level.

4.3 Turkish Speech Database Preparation

A Turkish speech database was prepared as part of this PhD study. The Turkish database[3] is available through the Evaluations and Language resources Distribution Agency (ELDA[4]).

The database includes read speech from 43 (22 males, 21 females) native speakers. The texts are selected from television programs and newspaper articles. The selection criterion was to cover as many different topics as possible. The phonetic balance of the text was also important for the selection. Selection of different topics helped to reduce prosodic effects in the database since speakers couldn't adapt to the topic due to rapid change.

There are two types of data in the database, continuous speech and isolated speech data. The isolated words are selected among most frequently used words [52]. The corpus statistics are shown in Table 4.2.

The data recordings are realized in quite room conditions. Average recording time for speaker was about 25 minutes, 22 minutes for continuous read speech and 3 minutes for isolated words. A small pause is inserted between each utterance in continuous speech and

[3] http://www.elda.fr/catalogue/en/speech/S0121.html
[4] http://www.elda.fr

Table 4.2: Turkish database corpus statistics.

Unit type	**Number**
Utterances(continuous speech)	215
Isolated words	100
Words in continuous speech	2160
Different words in continuous speech	1564
Different words in all recordings	1618
Male speakers	22
Female speakers	21

between each word in isolated words.

The recording materials were a portable Sony DAT-recorder TDC-8 and a close talking Sennheiser microphone MD-441-U. Speech was digitally recorded at 32 kHz sampling rate into DAT-recorder tapes and transferred to the digital storage media (hard disk) by using a digital sound card. Speech was then down-sampled into 16 kHz sampling frequency which is a sampling rate frequently used in speech recognition applications.

The database includes phonetic transcriptions of some of the data which is used for training an initial speech recognition system used for phonetic labeling of all the database. The initial segmentations were realized manually for the isolated words speech data from 5 males and 5 females. Each utterance in the database have a speech data file. Orthographic and phonetic transcriptions of utterances and isolated words were also created. In phonetic transcriptions, the phoneme list in Table 4.1 is used.

4.4 Language Modeling for the Turkish Language

As stated earlier in this chapter, agglutinative properties of the Turkish language make word-based language modeling very difficult. Another important aspect that make difficult the language modeling for Turkish is semantically determined word order property. This results in "free" word order which makes useless the classical language modeling. Although Turkish is considered as having an order of Subject-Object-Verb (SOV), it is possible to see all six permutations of this order. [53] states that 48% of sentences in a 500 naturally-occurring utterance list, have SOV word order while 25% have SVO word order, 13% have OVS word order and 8% have OSV word order.

The word order is determined by the semantic of the sentence. The sentence initial position is associated with the topic, the immediately preverbal position is associated with the focus and the post-verbal position is related to the background information.

The problems with word-based language modeling can be solved by different techniques adapted for agglutinative languages. These techniques are:

1. Part-of-speech tagging-based language modeling,

2. Morphology-based language modeling,

3. Stem-ending decomposition-based language modeling.

POS tagging based language modeling is based on POS tag replacements of words in the corpus [54]. In this approach, before applying n-gram modeling technique, the words and the punctuation marks in the corpus text are tagged with their morphosyntactic tag. The language model is then applied to the tag sequences. This technique needs an efficient an unambiguous POS tagger which is difficult for Turkish.

Morphology based language modeling uses morphological structure of the language and decomposes words by using morphological tagging of words. This method is proposed by [48] for Turkish and by [55] for Czech language which is also a highly inflected language.

In morphology based language modeling, the language model probability of a word is estimated from its morphological composition. The word history is not used for this estimation but the morphological structures are used. The aim is to maximize the probability that the tag sequence T is occurred for a given word W:

$$p(W) = arg \max_{T} P(T|W) \tag{4.1}$$

where $p(W)$ is the maximum language model probability computed for different morphological parses of W. $P(T|W)$ is the probability that tag sequence T is observed for the word W. This equation can be rewritten from Bayes' formula as:

$$p(W) = arg \max_{T} \frac{P(T)P(W|T)}{P(W)} \tag{4.2}$$

The probability $P(W)$ in the right hand side of this equation can be ignored by the assumption of "all words have equal chance of being appeared". T includes the root word and all morphosyntactic features to determine the word which means:

$$P(W|T) = 1 \tag{4.3}$$

The tag sequence T can determine the word W. From this property, we can rewrite the equation (4.2) as:

$$p(W) = arg \max_{T} P(T) \tag{4.4}$$

The trigram tag model is then used for determining the probability $P(T)$.

$$P(T) = \prod_{i=1}^{n} P(t_i|t_{i-2}, t_{i-1}) \tag{4.5}$$

where

$$P(t_1|t_{-1}, t_0) = P(t_1)$$

$$P(t_2|t_0, t_1) = P(t_2|t_1)$$

Each t is a morphological structure and can contain one root word and a number of inflectional groups (IG). An example for IGs:

sağlam+laş+tır+mak
sağlam+Adj^ DB+Verb+Become^ DB+Verb+ Caus+Pos^ DB+Noun+Inf+A3sg+Pnon+Nom
to cause (something) to become strong / to strengthen (something)

The inflectional groups in the example above are named according to their function. ^ DB is the derivational boundary. The inflectional groups that will be used in language model could be:

1. sağlam
2. Adj
3. Verb+Become
4. Verb+Caus+Pos
5. Noun+Inf+A3sg+Pnon+Nom

+Become:become verb, +Caus: causative verb, +Pos: positive polarity, +Inf: infinitive form of the verb, A3sg: 3rd singular person agreement, +Pnon: no possessive agreement, +Nom: nominative case.

It is possible also to define a second language model which includes only root words. The number of distinct root words in language model can be as low as 24,000 which is assumed to be sufficient for the morphology tagger of [49]. The two models can then be used in conjunction to estimate a language model probability for the word hypothesis obtained from the acoustic model.

Morphology based language modeling used in [55] is slightly different then the language modeling explained above which is designed for the inflectional structure of the Turkish language. Instead of using a tag sequence, a word is decomposed into two portions called stems and endings. This decomposition is obtained through a morphological analyzer for the Czech language.

Stem-ending decomposition based language modeling for the Turkish language, proposed in [58] and [57], is a modified version of morpheme based language modeling for the Czech language introduced by [56]. The original modeling procedure is summarized as follows:

1. Identify all possible endings for the language by using a vocabulary.
2. Extract the endings from all dictionary words. This can be done either by using a dictionary in which endings and stems are already defined or by processing the text to find endings and stems.
3. Take a sufficiently large text and by using the method in step 2, generate a text composed of stems and endings separated by white spaces.

4. Construct the vocabulary to be used for language model from the text generated in step 3.

5. For each stem, calculate a set of probability for the endings.

6. generate an n-gram language model for any combination of stems and endings.

A new modified version of this method is defined here for the Turkish language which has a different inflection characteristic from Czech language which is a Slavic language. The degree of inflectional and derivational suffixation in Turkish is much higher. The modification applied to the language modeling above are:

- Instead of determining all possible endings in step 1, they are extracted from a text corpus with the help of a morphological parser, [49].

- In step 2 the morphologically parsed corpus is used to extract endings and stems from the corpus. In the morphological parse, the root words are taken as stem and the remaining part of the word is taken as ending.

- The step 3 is applied by by using the decomposition process based on morphological parse.

The decomposition process can be explained by following examples:

The output of the morphological parser can be:
abartabilir @ *abart* +Verb+Pos+Verb+Able+Aor+A3sg
- (he can exaggerate)
abartacak @ *abart* +Verb+Pos+Adj+FutPart+Pnon
- (-something- to be exaggerated)
abartacak @ *abart* +Verb+Pos+Fut+A3sg
- (he will exaggerate)
abartan @ *abart* +Verb+Pos+Adj+PresPart
- (-the person- who exaggerates)
abartanlar @ *abart* +Verb+Pos+Adj+PresPart+Noun+Zero+A3pl+Pnon+Nom
- (-the persons- who exaggerate)

This parse is then used to generate the following decomposition:

abart + abilir
abart + acak
abart + an
abart + anlar

The new text which will be used for n-gram language modeling is composed of stems an endings. Since there will be common endings, the vocabulary will not increase rapidly and the number of OOV words that don't exist in the corpus may also be modeled thanks to this decomposition.

4.5 Experiments and Results

Acoustic models of Turkish speech recognition are obtained from Turkish continuous speech database prepared as a part of the doctorate study. The Speech Training and Recognition Unified Tool (STRUT) [86] developed at Signal Processing and Circuit Theory (TCTS) department of the Faculté Polytechnique de Mons (FPMs) is used for experiments.

STRUT include separate programs for different steps of speech recognition process which are grouped as front-end, training, decoding steps. Different techniques for each step are implemented in the tool.

STRUT programs are used for obtaining Turkish acoustic models as follows:

1. Create an acoustic model for phonetically segmented data.
2. Train the acoustic model with given data.
3. Recognize the training data and make an alignment of phoneme boundaries by using the trained model.
4. Train again for improving the accuracy of the model according to new phonetic segmentation which should be better than manual segmentation of step 1.
5. Add new unsegmented data.
6. Make phonetic segmentation of new data by the initial acoustic model obtained in step 4.
7. Train the model again with new segmentation to improve the accuracy for new data.
8. Align phoneme boundaries by new model of step 7 and train again until no more increase in the cross validation rate of MLP.

The feature vectors in the front-end are RASTA [8] features. The initial phoneme HMMs are 3-state HMMs with an initial state a middle state for the phoneme and a final state. There is a self loop for the middle state with the probability of 0.5.

The MLP has 3 layers, input layer, hidden layer and output layer. Input layer contain 9 frames of acoustic data which is useful for modeling the context of acoustic frames by the neural network. Hidden layer is composed of 1000 neurons. The size of hidden layer can be

Table 4.3: Turkish speech recognition results.

Test	**Correct %**	**Substitutions %**	**Deletion %**	**Insertion %**
Test 1	57.9	40.4	1.7	16.4
Test 2	60.0	38.3	1.7	15.3
Test 3	99.3	0.6	0.1	0.5
Test 4	87.9			
Test 5	75.4			

modified according to the processing time limits or the data available. Higher hidden layer sizes result in high processing time. When the amount of data is insufficient, the hidden layer must be kept small. The output layer of MLP is composed of the phonemes. There is one output for each phoneme. The output layer size is 30 for the Turkish. The output layer provides emission probabilities for middle state of phoneme HMMs.

There are four recognition tests listed below for the test data defined on Turkish database, one test for isolated word recognition and three tests for continuous speech recognition. There is no mismatch between the test and training conditions. Tests are defined as follows:

1. Continuous speech recognition with the vocabulary of the database (1618 words). Test set includes 215 sentences from 3 speakers.

2. Continuous speech recognition with the vocabulary of the database (1618 words). Test set includes 15 sentences from 40 speakers.

3. Continuous speech recognition with the vocabulary of the database (1618 words) by using a special grammar which include 215 sentences found in the test data and 215 sentences obtained by randomly deleting one word from each utterance. Test set is the same as in the test 2.

4. Isolated word recognition with a vocabulary of 200 words which includes the 100 words found in the test data and 100 randomly selected word from the vocabulary of the database.

5. Isolated word recognition with the vocabulary of the database (1618 words).

Table 4.3 shows the results obtained from the tests. When there are no constraints in the grammar, the error rates are high because of existence of similar words. But when the number of acceptable hypothesis is reduced as in test 3, the results are good enough. The isolated word recognition task of test 4 show that the accuracy is about 90% which is an acceptable level since the vocabulary includes similar words and small words. When the number of competing word hypotheses is high as in test 5, the accuracy decreases. One can test the system for name recognition and the resulting accuracy of the system would be higher since the competing hypotheses will be different as it is the case in test 3.

Table 4.4: Turkish language modeling results.

Test	**Perplexity**	**OOV Rate %**	**3-gram hits %**	**2-gram hits %**	**1-gram hits %**	**Entropy**
Test 1	388.96	6.86	52.57	29.41	18.02	8.60
Test 2	96.97	2.87	68.81	24.41	6.78	6.60
Test 3	108.54	1.47	66.66	25.20	8.14	6.76

The other modeling technique used in speech recognition is language modeling which is explained in section 4 of this chapter. Efficiency of the language modeling technique introduced in this thesis is tested on a large text and the results are interpreted for perplexity, OOV rates and rate of n-gram hits.

The two language modeling techniques are compared in the tests below:

1. Classical n-gram language modeling which is based on word histories.

2. New language modeling technique defined for agglutinative languages. The words are decomposed in stems and endings and the new vocabulary contains the stems and endings obtained from decomposition.

For training of the language models, the statistical language modeling toolkit developed at Carnegie Mellon University (CMU SLM) [87] is used. A training text of approximately 10 million words collected from online newspapers is used for training language models. Three language models are defined as follows:

1. Language model with a vocabulary of 60.000 words and classical n-gram modeling of original words in the training text.

2. The words in the training text are decomposed into stems and endings as it is explained in section 4 of this chapter, then an n-gram language model is trained on the text containing stems and endings. The vocabulary of test 1 is decomposed and used in this test.

3. The language modeling technique is same as for test 2 but the vocabulary is changed. The most frequent 60.000 stems and endings from the decomposed training text are selected as vocabulary.

The results are listed in table 4.4. The table shows the results obtained from the language models created by the CMU SLM. The test set included 1 million words collected from online newspapers.

We can clearly see that the new language modeling technique reduced the perplexity and the OOV rate significantly. This is an expected result because decomposing words into stems and endings let the language model to model unseen words since it is highly probable that the

stem and the ending of new word is already in the language model. Even though the word creation is unlimited in agglutinative languages, the number of stems and endings remain small. The OOV words can be observed only for the proper names.

Chapter 5

Confidence Measures for Speech Recognition

Confidence measures are defined as posterior probabilities of correctness of a statistical hypothesis. Confidence measures for speech recognition are used to make speech recognition usable in real life applications.

This chapter is focused on the use of confidence measures for different tasks of speech recognition. Some new approaches and results are given.

The need for confidence measures comes from recent advances in speech recognition field that make possible for the use of systems in human-computer interaction applications. The sources of variability that may influence the input speech may result in poor recognition results. Detection of these problems may help users of speech recognition systems to avoid them and make the recognition process more performant.

The idea of confidence measures comes from significance tests for statistical estimations. In [61] it was stated that a good method for improving recognition accuracy of a speech recognition system is to reject hypotheses that fail to satisfy certain criteria. Confidence measure is such a criterion that gives an idea about the role of chance in the result. Error rate is of course a good measure for the performance of a speech recognizer but it does not give further information.

Confidence measures are used to give scores to the output of the speech recognizer. Sometimes they are used to make a decision about the correctness of the recognizer output. In this case, they can be considered as a test statistic used in a hypothesis test, the acceptance or rejection of hypothesized output is based on a pre-determined threshold confidence measure.

The confidence measure problem can be seen as a process of statistical hypothesis testing in which we want to decide either to accept or to reject the most likely word sequence provided by the speech recognizer. The acceptance or rejection is determined by a threshold confidence level. The values on one side of the threshold are rejected while those of the other side are accepted. The two types of errors that can occur are, as in hypothesis testing:

- Type I errors, when the hypothesis is rejected while it is true, it is also called *false rejection error* (FR),
- Type II errors, when the hypothesis is accepted while it is false, it is also called *false acceptance error* (FA).

From these two errors, we define the unconditional classification error rate (CER) as the metric used to evaluate the hypothesis testing:

$$CER = \frac{N_{type\ I\ errors} + N_{type\ II\ errors}}{N_{tested\ hypotheses}} \tag{5.1}$$

where $N_{type\ I\ errors}$ is the number of false rejections, $N_{type\ II\ errors}$ is the number of false acceptances and $N_{tested\ hypotheses}$ is the number of all tested hypotheses.

The definition of CER in equation (5.1) depends on the overall performance of the speech recognition system. This property influences the objectivity of the CER and the judgment on the efficiency of the confidence measure which is based on the CER. This problem about CER is resolved by defining test conditions for which the word error rate is set to 50%. This issue will be better explained in the Section 6.5.

The research on robustness of speech recognition systems is closely related to the confidence measures. Some adaptation procedures for the speech recognition system can be automated by using a good confidence measure that classifies adaptation data.

The next two chapters are dealing with the use of confidence measures for speaker recognition and speaker model adaptation.

In speech recognition, confidence measures are classified into two groups according to the type of model used for the measures:

- Acoustic model based confidence measures,
- Language model based confidence measures.

As it can be inferred from names, acoustic model based confidence measures use the outputs of acoustic models while language model based confidence measures use the language model probabilities. These two types are then combined to obtain an overall confidence measure for the output of speech recognizer. In some applications, only one type of confidence measure is used.

5.1 Acoustic Model Based Confidence Measures

Acoustic model based confidence measures use acoustic models to determine the level of correctness of the output words or word sequence.

The main sources of errors in speech recognition comes from poor modeling of acoustic data due to variabilities in speech recognition environment. These variability sources include:

- Phonetic variabilities,
- Acoustic variabilities,
- Within-speaker variabilities,
- Across-speaker variabilities.

Phonetic variabilities include context dependent differences in the phonemes. The pronounciation of a phoneme can change with the surrounding phonemes, words and sentences.

Acoustic variabilities include the changes in the environment and in the acquisition material. Noisy environments as well as the use of different microphones are two examples. These variabilities are very hard to model accurately and generally they are the main source of errors.

Within-speaker variabilities include changes in the emotional state (stress, nervousness and/or happiness), speaking rate or voice quality of the speaker.

Across-speaker variabilities include changes in socio-linguistic background, dialect and in the vocal tract length and shape that can be different for each speaker.

In speech recognition, we try to model each of these variabilities. An ideal system should recognize the speech in all conditions. Since it is not an easy task to have a model of all the sources of variabilities, some methods are used to increase the robustness of speech recognizer in different conditions.

Acoustic confidence measure is one of the methods used to increase the robustness of a speech recognition system. The other methods are noise robustness and speaker adaptation [62], [63].

Another important source of errors in speech recognition is existence of Out-Of-Vocabulary (OOV) words in the test speech. This problem can be observed when the acoustic data for a word which is not modeled by speech recognizer is given to the system as input.

The confidence measures defined here are tested on both conditions of erroneous speech recognition. Mismatched conditions and existence of OOV words. Some confidence measures give better results in mismatched conditions while others give better results for OOV words.

All the acoustic confidence measures we have tested in this thesis are based on posterior probabilities provided by MLP [18]. These posterior probabilities are provided for each frame and are independent of the context. This property makes these probabilities useful for confidence measure since they provide a good information for the classification of a single frame of acoustic signal.

In the decoding phase of speech recognition, the Viterbi algorithm provides the word with highest score which is composed of the best state sequence:

$$W = \{q_k^1, ..., q_k^N\} \tag{5.2}$$

where N is the number of frames in the acoustic signal for the hypothesized word W and k is the index of HMM state. The basic confidence measure, posterior confidence measure (PCM), is defined as:

$$PCM(W) = \frac{1}{N}\sum_{n=1}^{N}\log(P(q_k^n|X^n)) \tag{5.3}$$

where $P(q_k^n|X^n)$ is the posterior probability of being in state q_k at time n for the acoustic vector X^n, and N is the number of frames in the acoustic signal of hypothesized word.

The procedure for computing PCM can be summarized as follows:

1. Find best HMM state sequence for the recognized word.
2. Take the posterior probability vectors for each frame from the MLP.
3. Apply equation (5.3).

The other acoustic confidence measures defined in this section are based on different normalization techniques applied to the basic PCM. Some new approaches and some combination of normalization techniques are introduced.

5.1.1 Relative Posterior Probability Confidence Measure (RPCM)

In standard posterior based confidence measure, PCM, all the frames have equal weighing factors. In acoustic model matching, during decoding process, it is possible to select a frame with a low probability because of the score maximization on overall state sequence. Consequently the state path include low posterior probabilities for certain acoustic frames. In RPCM, the posterior probability of each frame is normalized by the highest posterior probability attributed for the frame. The average posterior probability after the normalization is then used as confidence measure. The RPCM is defined as:

$$RPCM(W) = \frac{1}{N}\sum_{n=1}^{N}\log\left(\frac{P(q_k^n|X^n)}{P(q_{best}^n|X^n)}\right) \tag{5.4}$$

where $P(q_k^n|X^n)$ is the posterior probability of being in state q_k for feature vector X^n, $P(q_{best}^n|X^n)$ is the best posterior probability for the current frame. Ideally these two probabilities must be same. Due to maximization of the overall probability during the Viterbi decoding process, it is possible to select a probability which is not the highest one but is a part of the state path leading to the highest overall score.

RPCM is computed as follows:

1. Find the best HMM state sequence for the recognized word.
2. Take the posterior probability vectors for each frame from the MLP.
3. Find the maximum posterior probability for each frame.
4. Apply equation (5.4).

5.1.2 Acoustic Prior Information Based Confidence Measures (PPCM)

The random source assumption for the acoustic data and the statistical nature of acoustic modeling can cause poor modeling of some phonemes. If we can obtain the information about how good each phoneme is represented in acoustic model during the training phase of speech recognizer, we will be able to define a confidence measure that uses that information to attribute a confidence level to the output of the recognizer.

The average posterior probabilities obtained from the acoustic model can be higher for certain phonemes while it is lower for the others. The idea is to compensate the lower average posterior probabilities by normalizing the posterior probabilities of speech recognition output when computing the confidence measure. The average posterior probabilities are computed as follows:

$$\overline{P}(q_k) = \frac{1}{T}\sum_{t=1}^{T} P(q_k^t|X^t) \tag{5.5}$$

where X^t represent the frames corresponding to the phoneme q_k as provided from the phonetic alignment; T is the number of observation for the phoneme q_k in the entire training set. Definitions are given for one state HMMs that means each phoneme is represented by one HMM state. $\overline{P}(q_k)$ is computed during training phase by the following procedure:

1. Find the best state sequence providing the known word sequence for the acoustic data of each utterance in training data. This process is commonly called phonetic alignment.

2. Find the average posterior probability provided by the MLP by using the state sequence in the first step.

The average posterior probability computed from equation (5.5) is the *acoustic prior information*. This information is used to define a confidence measure by the following formula:

$$PPCM(W) = \frac{1}{N}\sum_{n=1}^{N} \log\left(\frac{P(q_k^n|X^n)}{\overline{P}(q_k)}\right) \tag{5.6}$$

where $P(q_k^n|X^n)$ is the posterior probability of being in state q_k for the feature vector X^n, N is the number of frames in the current utterance.

5.1.3 Entropy Based Confidence Measure (ECM)

Entropy is a measure of uncertainty about the realization of a random variable. Entropy related to a discrete random variable X with the observations $\{o_1, o_2, ..., o_n\}$ and the probability distributions $\{p_1, p_2, ..., p_N\}$ is the measure of disorder for the variable. It is defined as:

$$H(X) = -\sum_{i=1}^{N} p_i \log p_i \tag{5.7}$$

In the context of confidence measures, the entropy is calculated for each frame and is independent of the optimal state sequence obtained after the Viterbi decoding process. Therefore, the entropy can be seen as a measure of adequacy for the acoustic model. When the entropy is high, the matching power of acoustic model is low. Entropy based confidence measure is defined as:

$$ECM(W) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} p(q_k^n|X^n)\log\left(p(q_k^n|X^n)\right) \tag{5.8}$$

where N is the length in frame of the utterance W, K is the number of phonemes modeled by acoustic model. The probabilities $p(q_k^n|X^n)$ are the posterior probabilities provided by MLP for each acoustic frame.

5.1.4 Phoneme Based Normalizations for Confidence Measures

Phoneme based normalizations can be applied to any posterior probability based confidence measure. In this approach, the normalization by the length of the word is applied in two steps:

1. Normalize the sum of posterior probabilities which are labeled by same phoneme over the length of phoneme in frames, to obtain a mean posterior probability for the current phoneme.

2. Normalize the sum of mean posterior probabilities for the phonemes of the word sequence by the length of word sequence in phonemes.

The phoneme based normalization process give the same importance to each phoneme in the word sequence regardless of the length in frames of the phonemes. The results obtained by applying phoneme based normalizations shows that this type of normalization leads to very good performances. The reason for this performance increase is that the poorly matched phonemes are kept as short as possible by the Viterbi decoding and during normalization on entire word, the effect of these phonemes is lost due to short durations attributed to them.

Phoneme based normalizations of different confidence measures are similar. For the basic confidence measure, PCM, phoneme normalization is defined as

$$PCM_{PN}(W) = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{n_s - n_e}\sum_{n=n_s}^{n_e}\log\left(P(q_k^n|X^n)\right), \tag{5.9}$$

where $P(q_k^n|X^n)$ is the posterior probability of being in state q_k for the feature vector X^n; M is the number of phonemes in the current word; n_s and n_e are the beginning and ending frame indexes of the current phoneme in the word.

Phoneme based normalization for other confidence measures can be defined in the same way. $RPCM_{PN}$ is defined as

$$RPCM_{PN}(W) = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{n_s - n_e}\sum_{n=n_s}^{n_e}\log\left(\frac{P(q_k^n|X^n)}{P(q_{best}^n|X^n}\right). \tag{5.10}$$

$PPCM_{PN}$ is defined as

$$PPCM_{PN}(W) = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{n_s - n_e}\sum_{n=n_s}^{n_e}\log\left(\frac{P(q_k^n|X^n)}{\overline{P}(q_k)}\right). \tag{5.11}$$

ECM_{PN} is defined as

$$PCM_{PN}(W) = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{n_s - n_e}\sum_{n=n_s}^{n_e}\sum_{k=1}^{K}p(q_k^n|X^n)\log\left(p(q_k^n|X^n)\right). \tag{5.12}$$

Phoneme based normalizations require a phonetic alignment and labeling of acoustic data. This alignment can be realized by the following procedure when the phonetic transcriptions of recognized word sequence is known.

1. For each frame, find the posterior probabilities from the MLP.
2. Read the phonetic transcription of the current word sequence.
3. If the posterior probability of next phoneme is smaller than that of current phoneme, label the current frame by the current phoneme; otherwise advance the current phoneme pointer by 1.
4. Repeat the procedure in steps 2 and 3 for all the frames.

Acoustic model based confidence measures are applied at the end of recognition process. This property makes them useful for making judgment on the recognition results. If the degree of confusability is high for the possible recognition hypothesis, the confidence measure will give smaller confidence levels. Acoustic confidence measures can be used to attribute scores to the output hypotheses when the recognizer provide an *n-best* hypothesis list.

5.2 Language Model Based Confidence Measures

Language model for a language can be seen as a confidence measure, since the information provided by the language model is the probability of observing the current word given the history. This probability give an idea about the correctness of decoded word sequence obtained from the acoustic model.

In [64] some language model based confidence measures are introduced. This section gives some modifications to these confidence measures by applying word based normalizations instead of phoneme based normalizations. Some weighing factors are also used as in [66] for unigram, bigram and trigram probabilities provided by the language model.

The basic language model based confidence measure, n-gram probability based confidence measure (NGCM), for a word sequence of D word can be defined as:

$$NGCM = \frac{1}{D}\log\left\{P(w_i|h)\right\}, \tag{5.13}$$

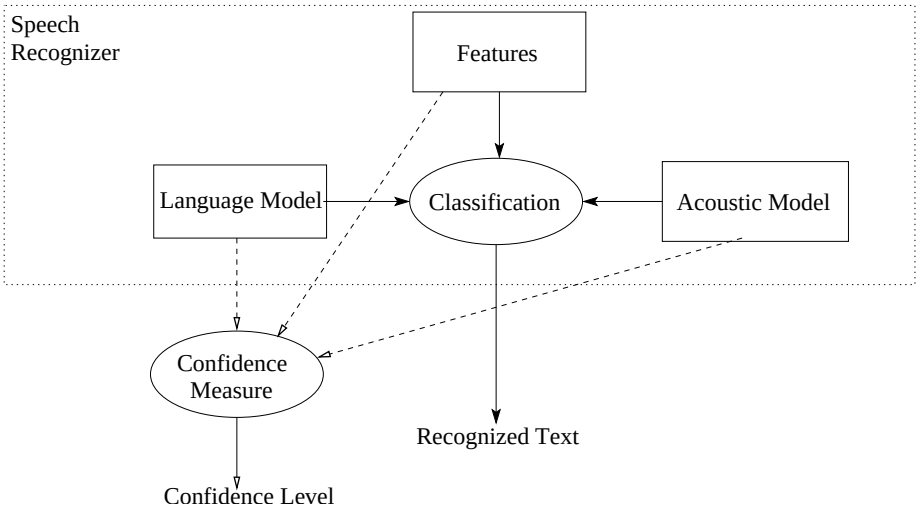

Figure 5.1: Use of confidence measure in speech recognition

where w_i is the current word in the word sequence at the output of the speech recognizer, h is the history.

Another interesting language model based confidence measure is use of a lattice and finding the density of this lattice for the entire word sequence. The lattice is constructed by language model probabilities. The most probable words for each time interval are used to create a lattice and the average number of words is used as the lattice density:

$$LD(n_s, n_e) = \frac{1}{N} \sum_{n=n_s}^{n_e} count(n) \tag{5.14}$$

where $count(n)$ is the function that computes the number of competing word hypotheses for the current frame. The competing word hypotheses are obtained by determining a threshold for the language model probability of each word in regard to its history.

Language model based confidence measures are applied in large vocabulary continuous speech recognition tasks while acoustic confidence measures can be applied in all types of speech recognition tasks. In this thesis we are interested in acoustic model based confidence measures.

5.3 Use of Confidence Measures in Speech Recognition

Confidence measures can be used for different purposes during or after the speech recognition process. The main goal is to mimic human listeners. When a human listener hears a word sequence, he/she automatically attributes a confidence level to the utterance; for example, when the noise level is high, the probability of confusion is high and a human listener will probably ask for repeat of the utterance. In Figure 5.1, the use of confidence measure in speech recognition is shown. The feature vectors and the acoustic model are used to obtain

posterior probabilities for each frame. Language model probabilities can be used for combined acoustic model and language model based confidence measures. There are two outputs of the new system, the confidence level and the recognized sequence. The confidence level is used to take further decisions on the recognized sequence.

The "confidence level" obtained from the confidence measure is then used for various validations of the speech recognition results. The main application areas of confidence measures are:

- Utterance verification,
- Keyword spotting,
- OOV detection, and
- Noise detection.

The use of confidence measure in these tasks can be compared to the test statistic that is used in hypothesis tests. The decision is sometimes based on the confidence levels obtained by one or more confidence measures [69].

5.3.1 Utterance Verification

Utterance verification is the process of verifying the result of a speech recognizer by using a confidence measure.

Confidence measures for the utterance verification task can be based on acoustic or language models. They are applied after the recognition of a whole utterance. It is possible to generate word graphs as in [68] which are formed only by the language model.

Combining several confidence measures generally performs better than the case where only a single confidence measure is used. Combined confidence measures are used for utterance verification task in [69]. The results show that there is some considerable improvement in classification accuracy when the vocabulary size is small.

Since utterance verification can be seen as a hypothesis testing problem, it is possible to use a test set to determine a threshold confidence level for the accept/reject decision.

5.3.2 Keyword Spotting

In keyword spotting, only acoustic model based confidence measures are useful because the number of keywords is limited and the history has no meaning for the keywords to be recognized.

Keyword spotting focuses only on recognizing the special keywords in the speech utterance instead of whole utterance. For example in the sentence "I want to speak to Mr. Brown", the keyword "Brown" is recognized and all the other words are ignored. This type of recognition

can be very useful in call centers since there are different types of forming a connection request. A keyword spotting system can also recognize correctly the sentence "Can I speak to Mr. Brown please", although the acoustic contents of two utterances are different.

As explained in Chapter 3, there are acoustic models known as "garbage" models in keyword spotting systems which are used to model the acoustic data not corresponding to any of the keywords. In the speech recognition process, if the acoustic data is matched by a garbage model, it is ignored.

Confidence measures for keyword spotting systems can be used to validate the decision taken by the recognizer for some acoustic data. In spoken dialogue management, the confidence level that is attributed to keywords may be realy helpful, for example to ease the dialogue and to improve efficiency [70].

Confidence levels can also be used in error handling in a dialogue systems as explained in [71]. The response can be generated as a function of confidence on the keywords. When the confidence level is low, the keyword can be rejected and the user is asked to repeat the utterance, when the confidence is medium, a confirmation of the keyword can be asked to the user, and when the confidence is high enough the keyword can be accepted and the next step in dialogue can be started.

5.3.3 OOV Detection

Detection of words that are not in the vocabulary of speech recognizer remains as an important problem of speech recognition systems since these words are tried to be recognized as the closest words in the vocabulary.

The most popular method to deal with OOV words is to use a garbage model which model all the speech not modeled with the word models. This technique does not give accurate result when the amount of OOV words is important during the speech recognition task. The garbage model can suffer from the generalization problem that leads to matching of in-vocabulary words.

Some confidence measures can detect high confusability of recognized words. For example the RPCM defined by the equation (5.4) will result in a low confidence measure when there is confusion.

OOV detection should be considered at word level. When the confidence measure is applied at utterance level, the effect of OOV word in the overall score could not be observed correctly. In [72], it is shown that word level likelihood ratio confidence measures which are computed by the ratio of likelihood probabilities of the two-best hypothesis, give good results for the utterance verification task when there are OOV words.

5.3.4 Noise Detection

Noise is one of important source of errors in speech recognition because it causes loss of important information related to the acoustic content of speech. Even if speech recognizer performs well in noise-free conditions, performance reduces rapidly in presence of noise. Noise robust speech recognition is one of the main research directions of todays' speech recognition. The problems with speech recognition in noisy environments are explained in [73], [74] and [75]. The methods explained in these works to deal with mismatched conditions are confirmed that they did not yield very good results, although provided some improvements.

Confidence measures can be used to evaluate the effect of noise in the recognition results by measuring the efficiency of the recognition for each word and each utterance. It is also possible to filter the noise-only or too noisy parts of acoustic data. A successful implementation of confidence measure based noisy acoustic data detection can be found on [76].

As it will be shown in the section 6.5, some confidence measures give better results for detection of noise than the others.

Detection of noise and provisionof a confidence measure, which is closely related to existence of noise level in speech, is very useful in adaptation of acoustic models to new conditions. When the noise level is too high, adaptation may be harmful for the performance of acoustic models.

5.4 Databases

The evaluation of confidence measures is realized on the Phonebook database [77]. Phonebook is a phonetically rich isolated word telephone speech database whose language is English. It was collected from 1300 speakers each reading 75 words over telephone lines. There are 8000 distinct words in the database.

The speech recognition system used for testing the confidence measures in this chapter is trained on the Phonebook database. The HMM/MLP hybrid speech recognition method is used. The test data set is selected to better observe the performances of confidence measures.

The best performing confidence measures are then tested on Turkish isolated word speech recognition task. Turkish speech recognition is based on the Turkish database prepared as a part of this thesis, which was made available through ELDA[1]. Isolated word and continuous speech test sets are used to test the efficiency of confidence measures on Turkish database.

5.5 Experiments

Three tests were defined on the Phonebook database in such a way that the initial recognition accuracy is set to 50%. The phoneme level normalization based confidence measures

[1] http://www.elda.fr

and the word level normalization based confidence measures are tested on these test sets. The test set definition procedure is as follows

1. Tests for noise effect. Noise is added to some data to cause recognition errors. The 50% recognition error is caused by noise which means all the utterances are correctly recognized before adding noise. The resulting data set is used for confidence measure tests.

2. OOV tests. 2000 correctly recognized utterances are selected from the test sets of Phonebook database. Confidence measures for the first and second hypotheses in the N-best list of the recognizer for each utterance are computed. All the confidence levels obtained for the first and second best hypotheses are used to plot the CER plots which show the performances of confidence measures. This test is considered as OOV test because when the word in the first hypothesis is deleted from the vocabulary of the recognizer, the second best hypothesis will be the output. Phone based normalizations for second best hypothesis are realized after phonetic re-alignment of acoustic data.

3. Clean data tests. The performance of recognizer is reduced to 50% accuracy by taking the words that caused recognition errors and randomly selecting equal number of correctly recognized words.

The results obtained for these three test sets are shown in next section with some discussions on the results.

The confidence measure tests on Turkish database are realized on two type of test sets:

1. Isolated words test set include 100 words from 43 speakers. Recognizer vocabulary is first set to 200 words including the words in the data set and 100 additional words. As a second test, 50% of words in the data set are deleted from the vocabulary to test OOV performances of confidence measures.

2. Continuous speech test set include 200 sentences from 3 speakers and 15 sentences from 40 speakers. Recognizer is forced to recognize sentences from a predefined list of sentences. The list is obtained by taking 215 sentences of test sets and 215 additional sentences obtained by deleting randomly one word from each sentence are added to the list. Two tests are realized, one with recognition of entire list of 430 sentences, one with only the 215 sentences obtained by deleting one word from each sentence.

Language model based confidence measures defined in Chapter 4 can be applied for the second type of tests. In next section we will give only the performances of acoustic model based confidence measures. Combination of acoustic confidence measures with language model based confidence measure of [66] remains to be investigated.

5.6 Sigmoid Matching

The confidence scores computed as above must be used to take the final decision of accepting or rejecting a hypothesis. Of course, we would like to have a value that can be directly interpretable so that the decision threshold can be easily fixed. A smart interpretation of such a value could be the probability of a word to be correct. Indeed, in such a case, a confidence score of 0.8 would mean that the word is statistically correctly recognized with 80% chance. During the training phase, we can build the histogram of word recognition rate according to their confidence score. We propose to match a sigmoid on this histogram. This sigmoid can be interpreted as a mapping function from the raw confidence score to probability-like values.

The procedure can be described as follows:

- for each confidence score [2], compute the word recognition rate as the ratio of the number of correct word on the total number of word, that is for each score i:

$$score(i) = \frac{h_{correct}(i)}{h_{correct}(i) + h_{incorrect}(i)}$$

- The sigmoid to be matched is as follows:

$$y = \frac{1}{1 + e^{-\beta(x-\alpha)}}$$

- For $(x - \alpha) = 0$ we find $y = 0.5$. This point can either be immediately taken from the histogram or preferably computed from the distributions of correct and incorrect words. Indeed, if we assume these distributions can be approximated by gaussians, we can find α as the point where the probability of a word to be correct is equal to the probability to be incorrect.

$$\alpha = \frac{\mu_{correct} * \sigma_{incorrect} + \mu_{incorrect} * \sigma_{correct}}{\sigma_{correct} + \sigma_{incorrect}}$$

  Where μ and σ are the mean and standard deviation of the gaussian distributions.

- The last unknown parameter is β which can be approximated by *golden section search* algorithm [67]. This algorithm finds a polynomial interpolation for a function that minimize an criteria. In our case, we want to minimize the distance between the histogram points and the sigmoid.

[2] actually, confidence score interval of the histogram

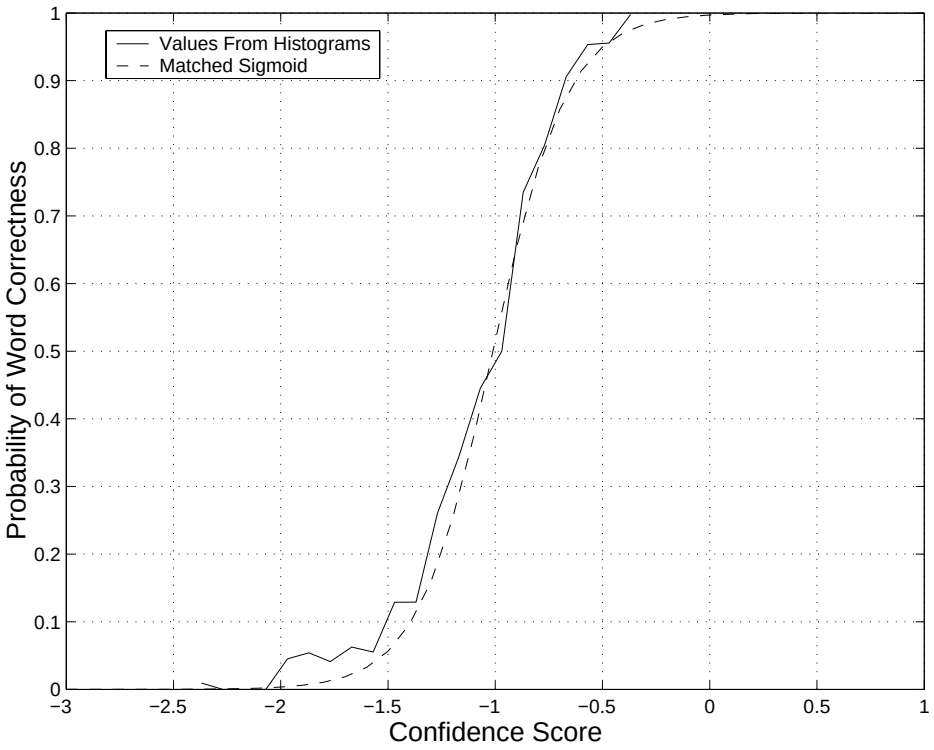

Figure 5.2: Mapping function (sigmoid) for the confidence measure $RPPCM_{PN}$ calculated over the test set prepared for `OOV` test

5.7 Results and Discussion

The results are presented in two parts. First part show the efficiency of different confidence measures for clean speech, noisy speech and for presence of OOV words in the speech. The tests realized in this part are based on Phonebook database which is an isolated word telephone-speech database as explained before. The second part show the performance of best confidence measures for Turkish isolated word and Turkish LVCSR tasks. The Turkish LVCSR task is restricted to test the efficiency of confidence measures when there are deletions.

Figure 5.3 and 5.4 show the effects of word level and phoneme level normalizations for different confidence measures when the source of errors in the recognition is noise presence.

Figure 5.5 and 5.6 show the effects of word level and phoneme level normalizations for different confidence measures when the recognition errors are caused by OOV words.

Figure 5.7 and 5.8 show the effects of word level and phoneme level normalizations for different confidence measures applied on clean speech.

The CER plots of Figures 5.3 thru 5.8 are obtained by computing CER values as explained in the beginning of this chapter for changing threshold confidence levels. Type I and type II errors are computed for different thresholds. In the worst case, the CER is 50% which is the initially determined error rate for the test sets. The best case is observed when the curve has

the minimum CER value for the optimum threshold.

The results on the first part of experiments show that $PPCM$, the newly proposed confidence measure, outperforms all the other methods in noise presence. When OOV words are the principal sources of errors, the best performing confidence measure is $RPPCM$ obtained by combination of $RPCM$ and $PPCM$. This is an expected result because in OOV situations the best probability for each frame is not selected. When the best probability is used for normalizations, the resulting confidence level should be low.

From the experiments, it is shown that phone level normalizations improve the efficiency of confidence measures. For the normal test conditions of Figures 5.7 and 5.8 it was observed that the newly defined $PPCM$ performs better. We can conclude that when the OOV rate is not high, $PPCM$ has a good performance either in the case of word level normalization or in the case of phoneme level normalization.

Figure 5.9 and 5.10 show the histograms for Turkish isolated word recognitions when all the words are in the vocabulary and when there are 50% OOV words.

Figure 5.11 and 5.12 show the histograms for Turkish continuous speech recognitions for recognition of correct sentences and for the sentences with one deleted word.

The second part of the experiments can be seen as a verification of the conclusions of first part on Turkish database. The $PPCM_{PN}$ is used for the experiments. The results are shown as histograms of raw confidence scores. The raw confidence measures can be transformed to probability-like values by a sigmoidal function as explained in [65]. The transformation is not applied to the values since we are comparing two histograms. The interpretation of raw confidence scores is as follows:

- When the hypothesis has a high raw score, the confidence on the hypothesis is lower.
- When the score is low, the confidence on the hypothesis is higher.

If we apply these interpretations to the Figures 5.9 and 5.10, when the histogram bins are higher on the left side of the graphs, confidence on the decodings is higher; when the histogram bins of right side are higher we can conclude that decodings have a lower confidence.

In Figure 5.9, the confidence on decodings is higher which is an expected result since most of the hypotheses are correct, in Figure 5.10, the confidence is lower because existence of OOV words in the utterances caused high error rate. The results show the efficiency of confidence measure for detecting OOV errors.

In Figures 5.11 and 5.12, the two histograms are not separated as good as the histograms of isolated word recognition in previous test. The results show that the confidence measure is inefficient for this test. This is an expected result since deleting one word will not effect the confidence score of entire sentence which is long enough. The effect of low confidence scores attributed to the frames of deleted word is minimized over the long sentences.

In conclusion, we showed that newly defined acoustic model based confidence measure

is efficient for OOV word detection and utterance verification in noisy conditions. Further researches should be carried on to integrate language model based confidence measures.

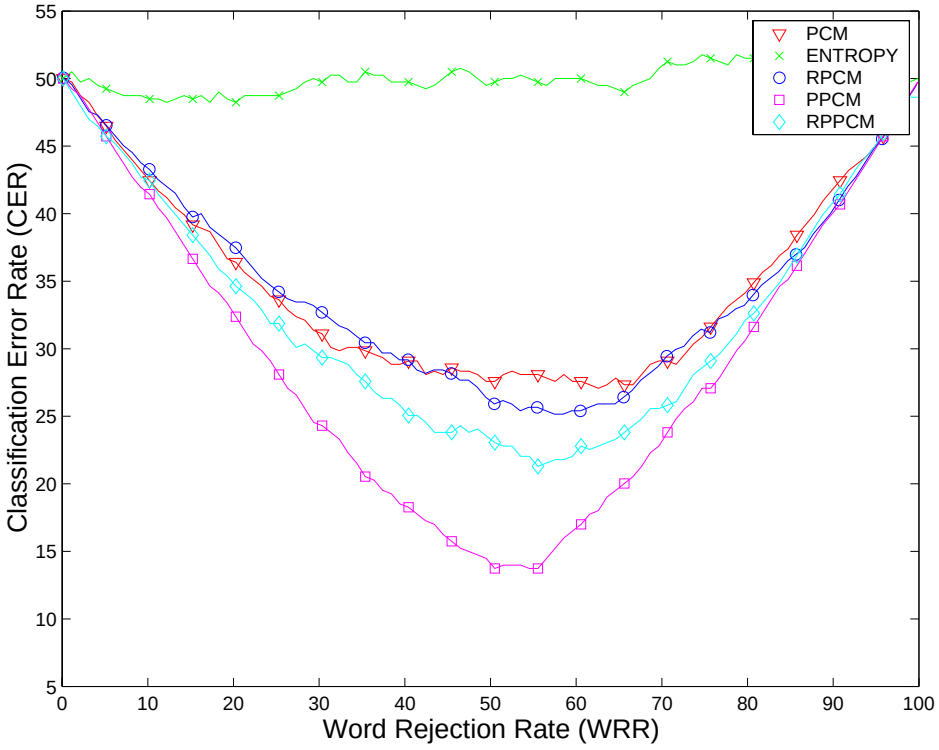

Figure 5.3: CER plot of word level normalization based confidence measures for noise effects.

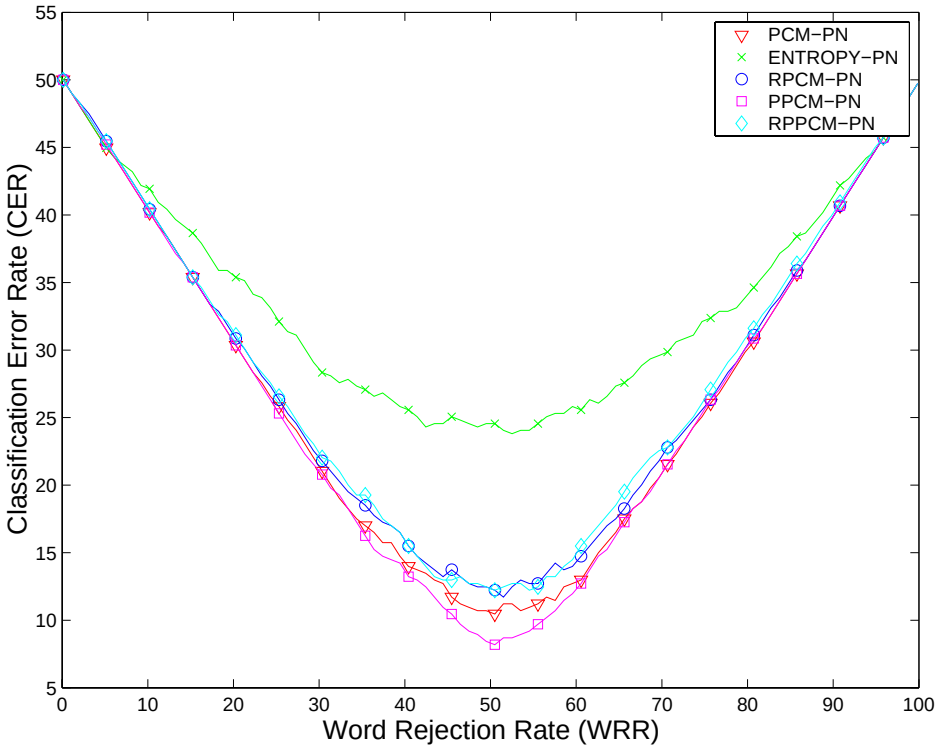

Figure 5.4: CER plot of phoneme level normalization based confidence measures for noise effects.

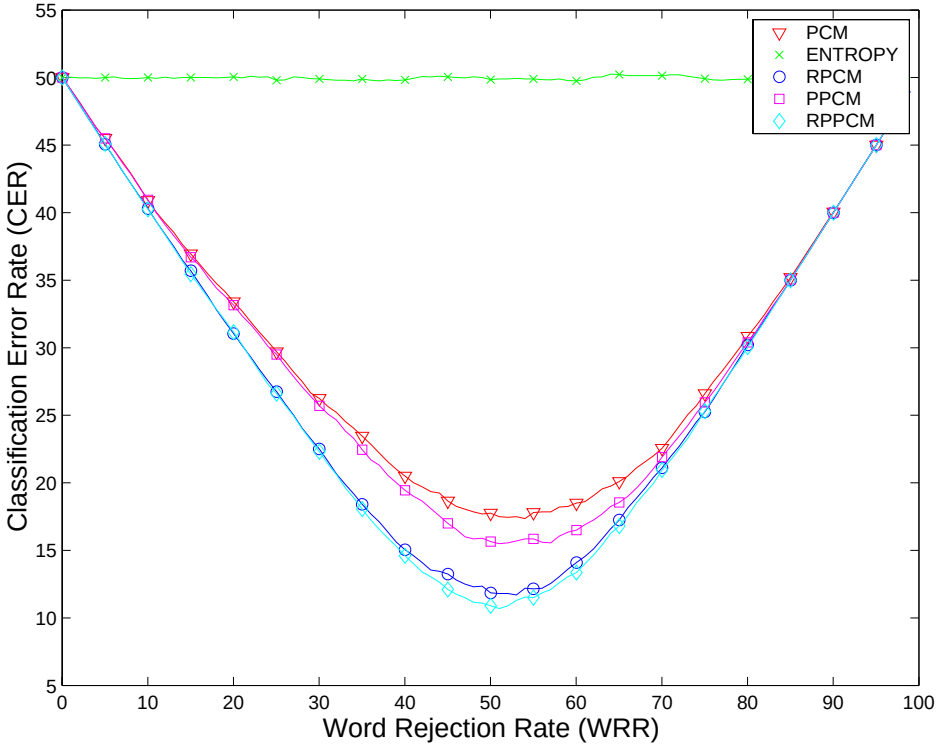

Figure 5.5: CER plot of word level normalization based confidence measures for OOV effects.

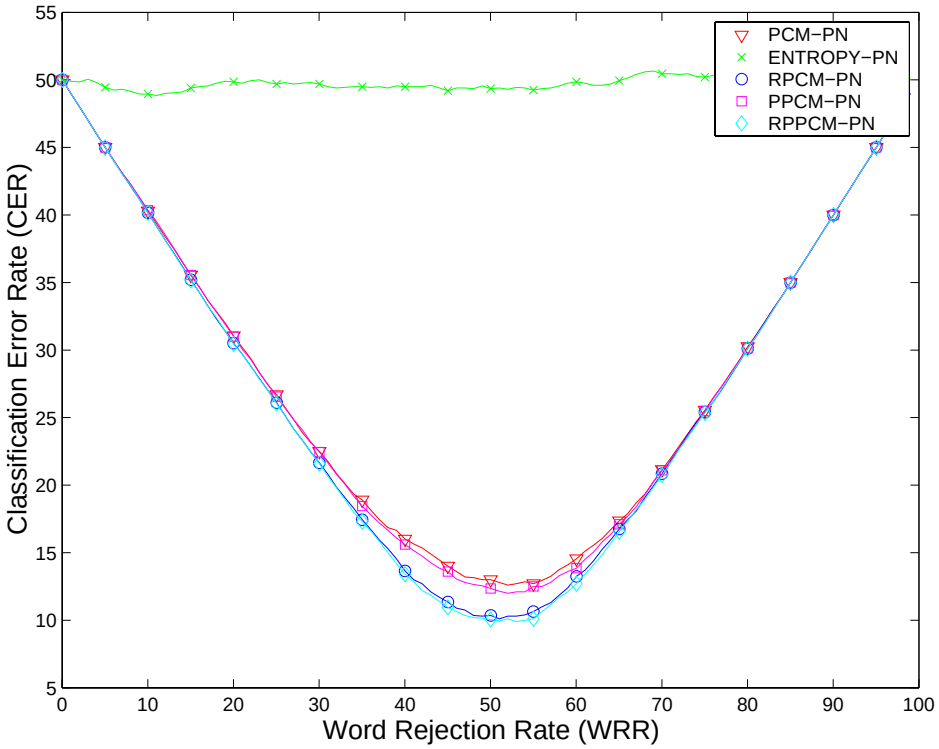

Figure 5.6: CER plot of phoneme level normalization based confidence measures for OOV effects.

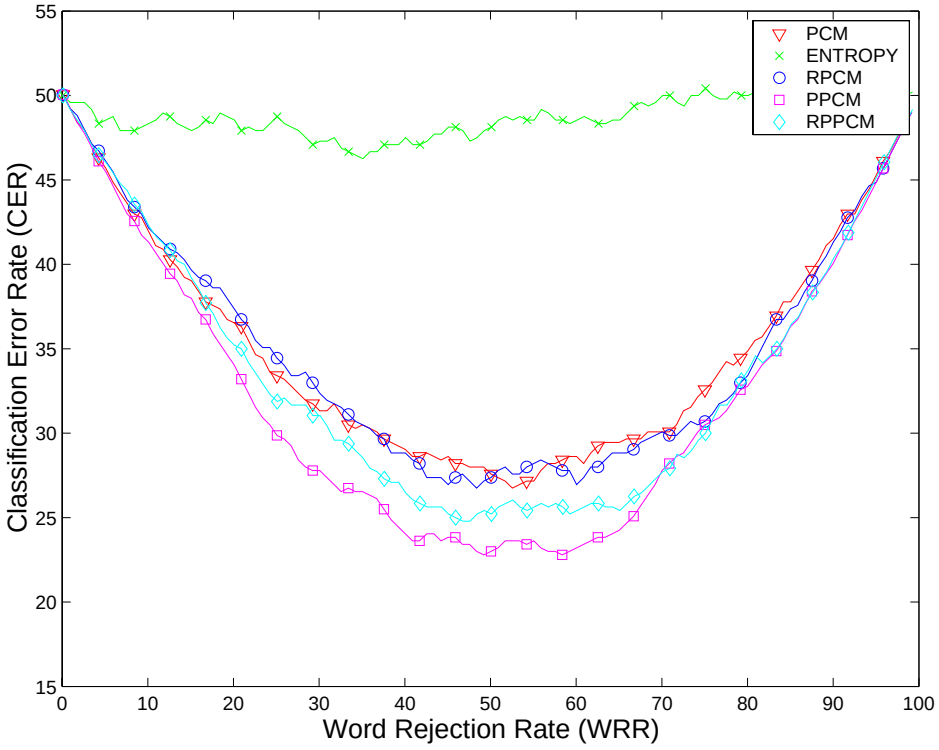

Figure 5.7: CER plot of word level normalization based confidence measures for clean speech.

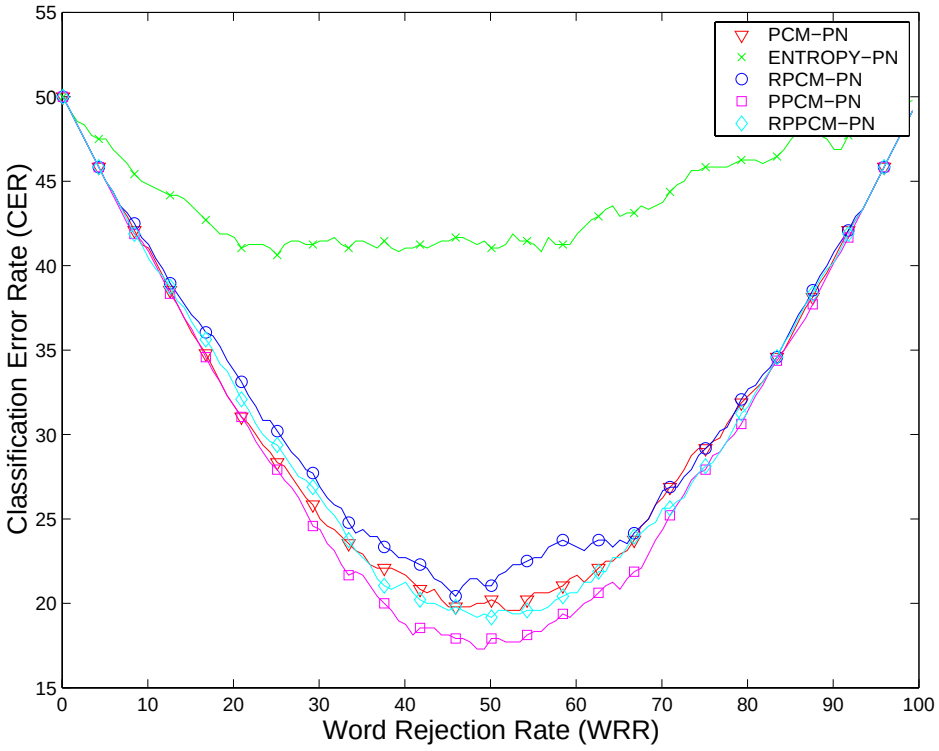

Figure 5.8: CER plot of phoneme level normalization based confidence measures for clean speech.

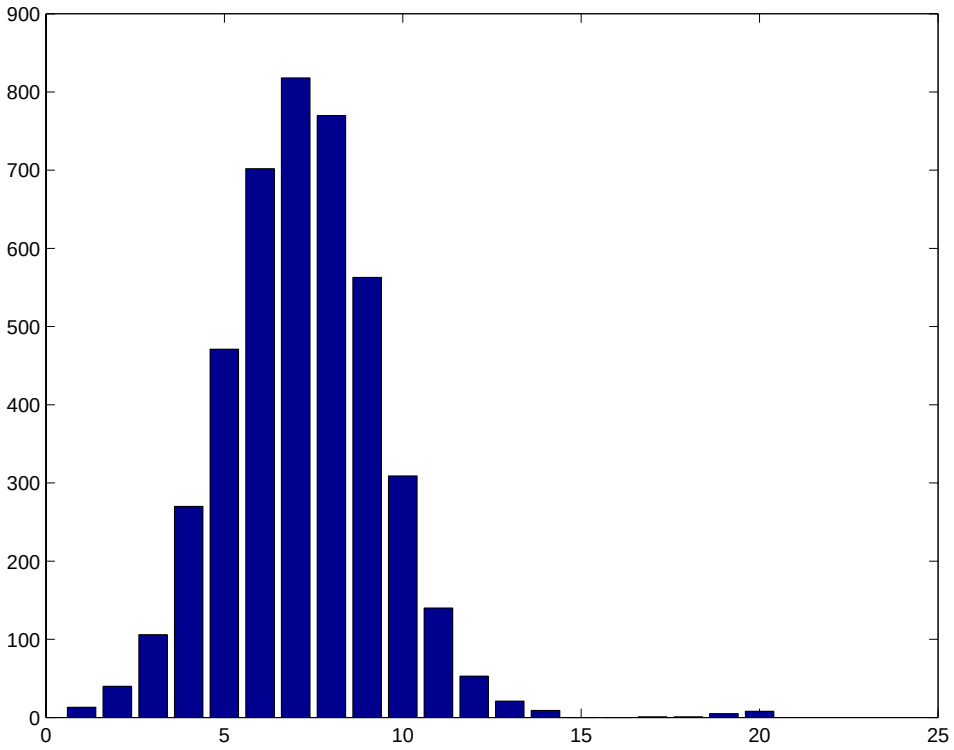

Figure 5.9: Histogram for confidence levels of isolated words when all the words are in the vocabulary.

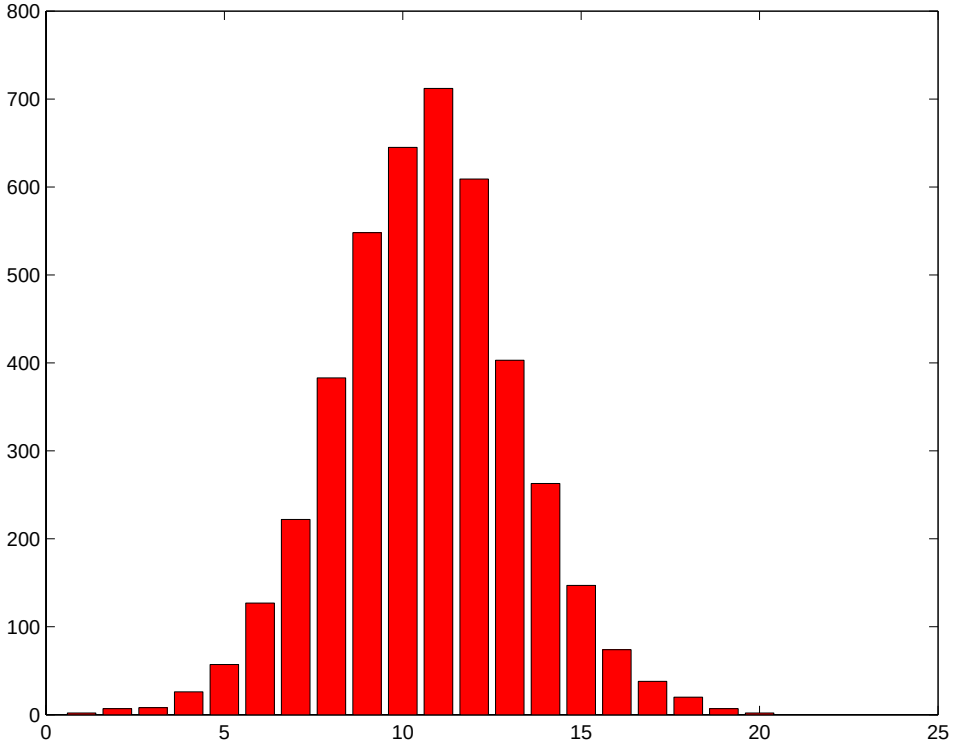

Figure 5.10: Histogram for confidence levels of isolated words when 50% of the words are OOV words.

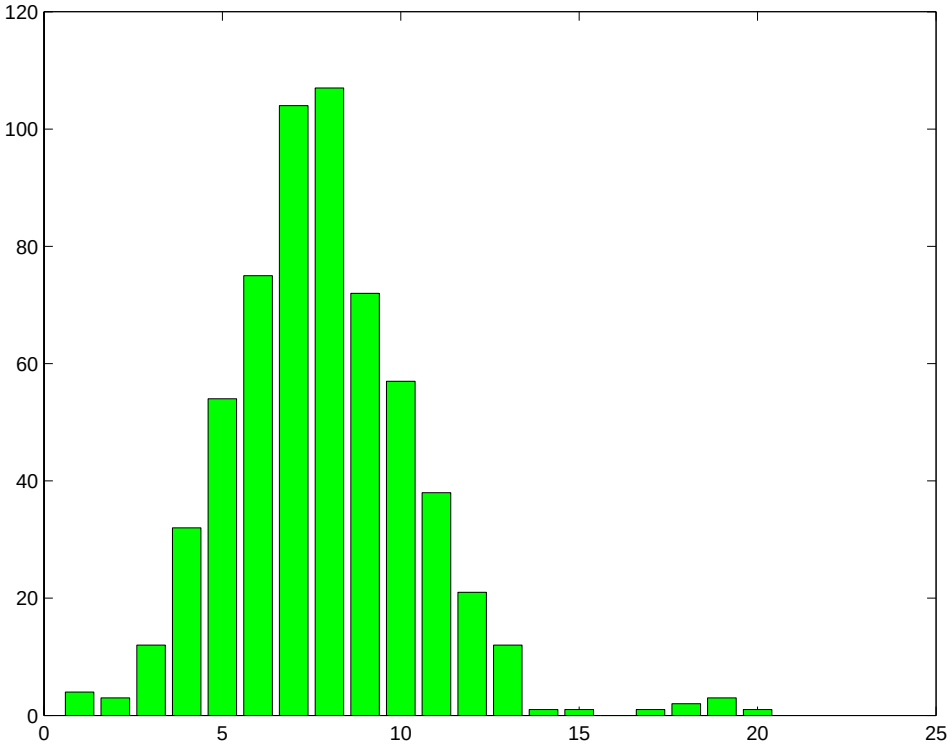

Figure 5.11: Histogram for confidence levels of sentences when they are correctly recognized.

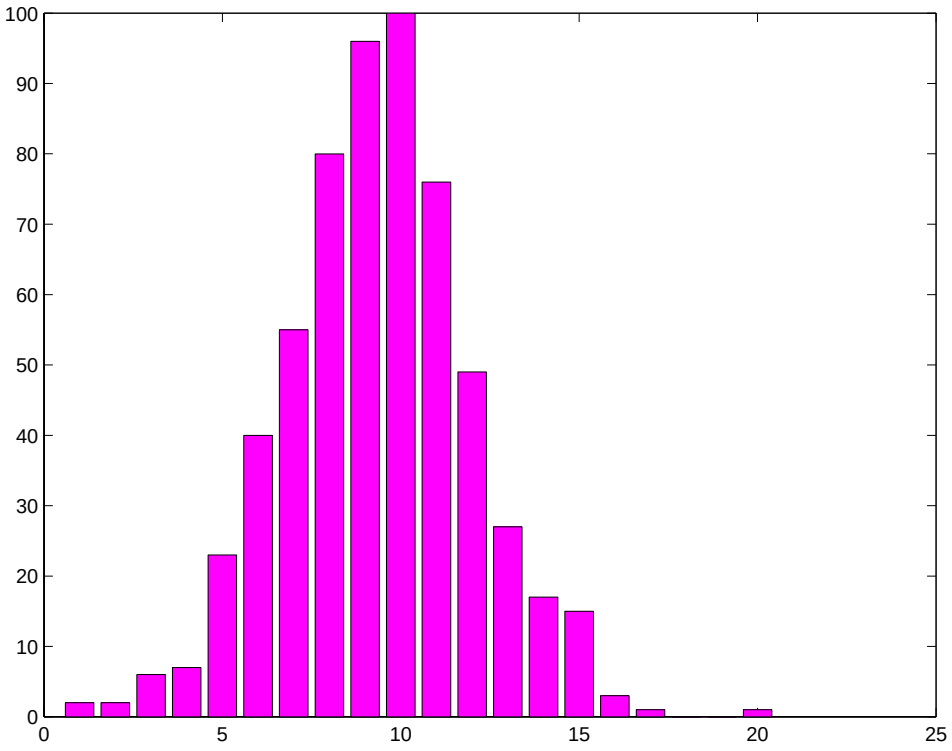

Figure 5.12: Histogram for confidence levels of sentences when one word is deleted from each sentences.

Chapter 6

Confidence Measures for Speaker Recognition

Confidence measures for speaker recognition are different than those used for speech recognition because the nature of the problem is different in these two types of acoustic data based recognition tasks.

Confidence measures for speaker recognition should be considered separately for speaker verification and speaker identification tasks.

Speaker identification can be compared to an isolated word speech recognition system. The difference is that when the identification is text independent, there is no phonetic transcriptions for the recognition results which means the advantages of phonetic labeling cannot be observed for speaker identification confidence measures.

The confidence measure experiments and the results obtained in this chapter are based on speaker verification systems. A new approach is introduced for interpretation of the confidence measures based on inverse Fisher transformation.

6.1 Confidence Measures for Speaker Identification

Speaker identification is based on training acoustic models for each speaker and also training some garbage models to match impostors.

As it is stated in [78], the voice characteristics of speakers could not be extracted and used in a deterministic approach. Voice differ from fingerprints and genetic imprints in several aspects:

- Voice changes over time, either in the short-term (days) or in the long-term (years).
- Emotional state of speaker can affect the voice.
- Voice can be altered voluntarily through imitation.

These properties of voice make it less usable when the security needs are high for example in forensic identification. However, easiness of identification by voice is an important advantage for voice based identification.

Good performances with speaker identification applications are obtained in certain conditions:

- Speaker must be cooperative with the system,
- Noise level in identification environment must be acceptable,
- Decision thresholds must be determined with sufficiently large data covering similar conditions as the conditions in the real environment.

The use of confidence measure provides a good interpretation of speaker identification results. The speakers with a low confidence levels can be asked for another try or can be rejected according to desired security level. Low confidence can occur in following situations:

- Change in the voice of the speaker. This situation will result in a false rejection of the speaker.
- Presence of noise or other environmental mismatches between test conditions and the training conditions when estimating the speaker models. This situation will result in either false rejection or false acceptance. The system behavior is hard to predict. Confidence measures can be useful for detection of this situation.
- Impostor access. If the decision thresholds for speakers are not determined well enough, impostors may be identified as one of known speakers which will result in a false acceptance. Confidence measures can detect impostor presence since the impostor will have a low confidence score.

Confidence measures for text dependent speaker identification are similar to those used for speech recognition. All of the acoustic confidence measures defined in chapter 5 can be applied directly to text dependent speaker identification including phoneme based normalizations.

For text independent speaker identification, phoneme normalizations are not usable. Only the posterior probability confidence measure (PCM) can be used in this case. When the number of speakers is high, the computational cost of this type of measure will be increased. The techniques used for speaker verification confidence measures can be applied successfully to text independent speaker identification when there are garbage models for each speaker.

6.2 Confidence Measures for Speaker Verification

Speaker verification offers a more accurate and efficient method than speaker identification. Data acquisition for speaker verification is easier than other biometric verification techniques like fingerprint verification or iris structure verification.

A typical speaker verification system includes three steps:

1. Enrollment or training step for the new users. In this step speaker models are created and trained with the data collected from each new speaker. The model is stored in the system for future usage in the verification process. Decision thresholds for each speaker are also determined in this step.

2. Verification of the speaker's voice. This step includes measuring a distance between the stored model of the claimed speaker and the speech data fed to the system. An accept or reject decision is taken by comparing the distance with a predetermined threshold for each speaker.

3. Adaptation of the speaker model to new conditions or to changes in the speaker voice. This step is used to improve the efficiency of the system when there are changes either in the verification environment or in the voice of a particular speaker. Environmental changes require adaptation of each speaker model.

There are several type of speaker verification techniques according to the restrictions applied to the speech required by the system. Speaker verification can be text dependent, text prompted or text independent. Confidence measures for the speaker verification can be different according to the type of the system. In this thesis we are focused on confidence measures for text independent speaker verification systems.

Confidence measures defined here for speaker verification are based on likelihood ratios obtained from a GMM based speaker verification system. GMM method is explained in chapter 3. GMM based speaker verification systems and adaptation methods for these systems will be detailed in the next chapter.

The likelihood ratio is defined as the ratio of likelihood score of the claimed speaker model for the input speech to the likelihood score of the world model. This ratio can be defined as:

$$\Lambda(X) = \frac{1}{N} \sum \log \left(\frac{p(X|\lambda_C)}{p(X|\lambda_{\overline{C}})} \right) \tag{6.1}$$

where N is the number of acoustic frames, X is the representation of feature vectors obtained for input speech, $p(X|\lambda_C)$ is the likelihood probability obtained from speaker GMM for feature vectors X and $p(X|\lambda_{\overline{C}})$ is the likelihood probability obtained from world GMM for feature vectors X.

The likelihood ratio obtained in equation (6.1) is used to make a decision on acceptance or rejection of the speaker in a GMM based speaker verification system.

In order to apply the confidence measure, some impostor data attributed to each speaker are included in the system. It is possible to define impostor data for a group of speaker. For example impostor data for female speaker and impostor data for male speaker can be used. Impostor data are used to compute the means and standard deviations for impostor

likelihood ratios before the verification process. The means and standard deviations are then used to compute confidence levels for each utterance during the verification process.

The confidence level obtained by the confidence measure defined above is then transformed to the correlation domain, through inverse Fisher z-transformation [60], in order to obtain a value which is easily interpretable by the user. This transformation is the subject of the next section.

6.3 Fisher z-Transformation for Confidence Measures

Fisher z-transformation is a method used to define a confidence interval for the estimated correlation value of a population. The estimation is realized by using a small sample and some hypothesis tests can be applied to verify the correctness of estimation. In [60] a more efficient method is used to evaluate the efficiency of a correlation value. The method is to transform the correlation values to a normal distribution domain for better comparison when the correlation values are close to 1. The transformation is defined as:

$$z = \frac{1}{2} \log \left(\frac{1+r}{1-r} \right) \tag{6.2}$$

where z is the transformed value of the correlation value r which is estimated for the samples. z has a Gaussian (normal) distribution and is used to:

- Compare the correlation values and find the degree of difference between two correlation values.
- Combine two different correlation values obtained for the same population on different samples.
- Find confidence intervals for the correlation values.

The inverse of the transformation explained above is used to obtain an interpretable confidence measure for speaker verification. The inversion of equation (6.2) is defined as:

$$r = \frac{e^{2z} - 1}{e^{2z} + 1} \tag{6.3}$$

The value z in the equation (6.3) is the normalized likelihood ratio obtained by equation (6.1). There are two types of likelihood ratios used in confidence measures; likelihood ratio for the claimed speaker and likelihood ratio for the impostor. The normalizations are realized as follows:

$$z_{speaker} = \frac{\Lambda(X) - \mu_{speaker}}{\sigma_{speaker}} \tag{6.4}$$

and

$$z_{impostor} = \frac{\Lambda(X) - \mu_{impostor}}{\sigma_{impostor}} \tag{6.5}$$

where z is the normalized value, $\Lambda(X)$ is the likelihood ratio obtained by equation (6.1), μ and σ are the mean and standard deviation of likelihood ratios for current speaker. These values are determined during the training phase for the data used to determine the decision threshold. This normalization provides the equivalent likelihood ratio for a standard normal distribution.

Likelihood ratio has a Gaussian distribution with μ and σ as the mean and standard deviation of the distribution since it is obtained by multiplications of likeliood scores which are obtained by Gaussian pdfs.

The two likelihood ratios represented by $z_{speaker}$ and $z_{impostor}$ are then transformed into correlation domain as:

$$r_{speaker} = \frac{e^{2z_{speaker}} - 1}{e^{2z_{speaker}} + 1} \tag{6.6}$$

$$r_{impostor} = \frac{e^{2z_{impostor}} - 1}{e^{2z_{impostor}} + 1} \tag{6.7}$$

The values $r_{speaker}$ and $r_{impostor}$ are then used two determine a confidence level for the decision taken for the input data.

$$CM = r_{speaker} - r_{impostor} \tag{6.8}$$

where CM is the final confidence level attributed to the decision. A negative confidence level means that the utterance comes from an impostor. If the utterance is coming from the true speaker, ideally, $r_{impostor}$ must have a value close to zero and the $r_{speaker}$ will determine the level of confidence on the decision. When the impostor data and speaker data are not well separated, the computed confidence level will have a value close to zero which means that the decision is taken by chance.

6.4 Experiments

This section give some details about the structure of the speaker verification system used for the experiments in this thesis. The system structures explained in this chapter are used then as the bases of speaker adaptation experiments in the next chapter. GMM based text independent speaker verification systems are used in the experiments.

The POLYCOST [84] speaker recognition database is used for experiments. This database is a telephone speech database which include read speech and spontaneous speech from speakers of different countries. The language of read speech is English, spontaneous speech is in the mother language of the speaker.

Speaker verification system, as a statistical model based system, has two phases. Training and testing phases. In training phases, the speaker model parameters are estimated from a training data set. The testing phase includes usage of system when it is operational.

For each speaker there are three data sets selected from the database:

1. Training data set includes training data for each speaker.
2. Test data set includes the data to be used for threshold determination for each speaker.
3. Verification data set includes data for each speaker to evaluate the confidence measure.

The data for training world model, female impostor and male impostor are selected from the remaining data after the selection of three data set above.

Speaker verification is based on speaker specific characteristics. It is important to make a speech/non-speech classification of data collected from speakers. This classification can be done in two ways; speech/silence classification and voiced/unvoiced classification.

6.4.1 Speech/Silence Modeling

Silence is the part of a utterance which does not convey any phonetic information. It can be present in any part of a speech utterance. When the silence presence in the speech data is high, the statistical speaker model will be effected negatively. Hence it is important to ignore silence in statistical speaker modeling to have good performances.

In the experiments we used model based speech/silence classification which use a speech recognition system for initial speech/silence segmentation of training data. The speech recognition system was trained on the Phonebook [77] database (see chapter 5) which is a telephone speech database like POLYCOST database. Training phase for speech/silence modeling based speaker verification is as follows:

1. Initial segmentation of data reserved for world model by a speech recognition system.
2. Determining the mixture sizes for speech and silence models according to data available for each part. For example if the 1/4 of the data is silence, the mixture sizes may be 32/128.
3. Training GMMs for speech and silence.
4. Use the models obtained in step (3) for segmentation of the speaker data.
5. Determine mixture sizes for speaker models. Speaker GMMs have smaller mixture sizes than the world models.
6. Train speech and silence GMMs for each speaker.

The GMMs obtained by the procedure above are two-state GMMs, one state for modeling silence and another state for speech segments. In testing phase, the likelihoods computations are based only on the speech parts of data. When the likelihood of silence state his higher for any acoustic frame, it is ignored for the overall likelihood computation.

6.4.2 Voiced/Unvoiced Modeling

Voiced/unvoiced modeling is more useful than speech/silence modeling in text independent speaker recognition tasks because the speaker related characteristics are conveyed by the voiced parts of speech. The characteristics of vocal tract and vocal cords are transfered only by the voiced part of the speech. Voiced/unvoiced modeling of the speech allows ignoring unvoiced parts and training GMMs only on the voiced parts.

Voiced/unvoiced classification can be obtained by simple computations based on fundamental frequency or energy [85]. In this thesis we used GMM based modeling of voiced and unvoiced parts of speech for simplicity of integration into GMM based speaker models. The voiced/unvoiced modeling is applied as follows:

1. As in speech/silence classification, use a speech recognition system to obtain an initial voiced/unvoiced segmentation of the data that will be used for world model training.
2. Determine the mixture size according to the availability of data for two models.
3. Train GMMs for voiced and unvoiced parts of speech data to obtain a two state world model.
4. Use voiced/unvoiced states of world model to classify speaker data.
5. Train two state GMMs for each speaker with appropriate mixture sizes.

Use of voiced/unvoiced modeling is same as speech/silence modeling. Unvoiced parts of speech are ignored during likelihood computations.

6.4.3 System Architecture

The speaker verification system used in the experiments uses world models for likelihood ratio computations.

The training phase of the system is shown in Figure 6.1. As it can be seen on the figure, training phase is composed of three steps:

1. Training world model with a sufficiently large data set.
2. Training speaker models for each speaker.
3. Determining thresholds for each speaker by using some data from each speaker and some generic impostor data.

Speech/silence modeling or voiced/unvoiced modeling can be applied by adding segmentations and one more state for each model.

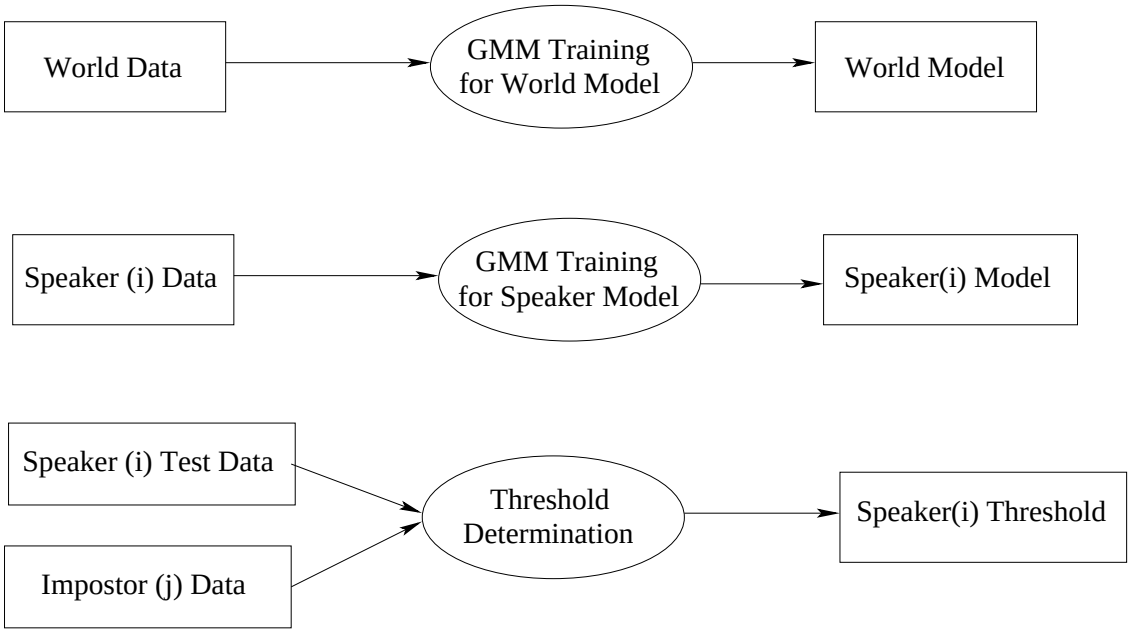

Figure 6.1: Training phase for the GMM based text independent speaker verification system.

Threshold determination in step(3) is based on equal error rates (EER) obtained by making verification tests. The EER is obtained when the rate of false rejections (FRR) and the rate of false acceptances (FAR) are equal. The FAR and FRR are computed as follows:

$$FRR = 100 \times \frac{Number\ of\ False\ Rejections}{Number\ of\ Tests\ for\ True\ Speaker} \tag{6.9}$$

$$FAR = 100 \times \frac{Number\ of\ False\ Acceptance}{Number\ of\ Impostor\ Tests} \tag{6.10}$$

$$EER = FAR = FRR \tag{6.11}$$

The likelihood score providing EER is selected as the decision threshold for the speaker. The impostor data used in step (3) of model training procedure above is divided into two group. The thresholds for male speakers are determined by using data from male impostors and for female speakers, female impostors data are used. Such grouping is based on assumption that the likelihood score provided by a model of male speaker, for the data coming from a female speaker should be low enough and including this data will cause erroneous FAR computations in equation (6.10).

Once the models are trained and the thresholds are fixed, the speaker verification system is ready to use. The confidence measures are applied in the verification phase. The mean and standard deviation parameters of equations (6.4) and (6.5) are computed for the likelihood ratios of speaker data and impostor data used in threshold the determination phase. The means and standard deviations are then used to compute confidence level for each utterance from the equations (6.4) through (6.8).

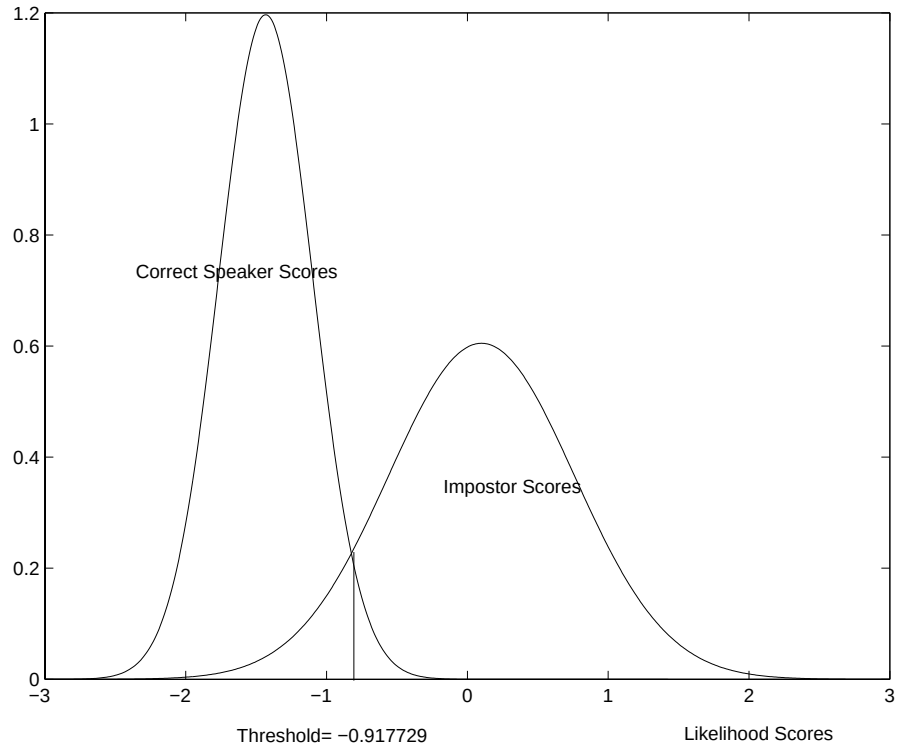

Figure 6.2: Scores obtained from a speaker model for the data from correct speaker and the data from impostor data.

6.5 Results and Discussion

Figure 6.2 shows estimated Gaussian distributions of likelihood ratios obtained by using a particular speaker model and the world model for the speaker data and the impostor data. When the two gaussians are well separated, the speaker verification system will be more efficient. The confidence measure for a likelihood score of speech data in the verifcation phase is a function of the distance of likelihood score from the means of two Gaussians. When the score is close to the mean of the true speaker Gaussian, the confidence level increases, when the score is close to the mean of the impostor Gaussian, the confidence level decreases. Confidence level is equal to "1" when the score is not between the two means, and is "0" when the score is equal to the threshold. For the true speaker side of threshold, the sign of confidence level is positive while it is negative for the impostor side.

Figure 6.3 shows the histogram of the confidence measures when the decision taken by the speaker verification model is correct for the speech data coming from correct speakers.

Figure 6.4 shows the histogram of the confidence measures when the decision taken by the speaker verification model is correct for the speech data coming from impostors.

Figure 6.5 shows the histogram of the confidence measures when the decision taken by the speaker verification model is incorrect for the speech data coming from correct speakers (false rejections).

Figure 6.6 shows the histogram of the confidence measures when the decision taken by the speaker verification model is incorrect for the speech data coming from impostors (false acceptance).

Figures 6.5 and 6.6 show that confidence levels for the verification errors are usually close to "0" which means the confidence measure is efficient.

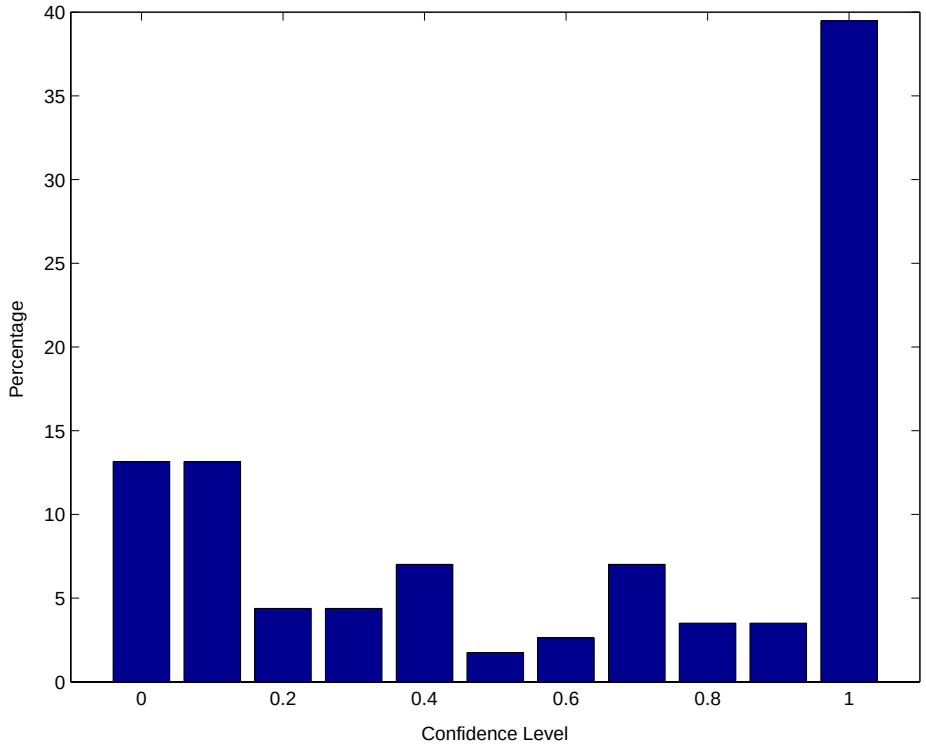

Figure 6.3: Efficiency of confidence measures for correctly accepted speakers.

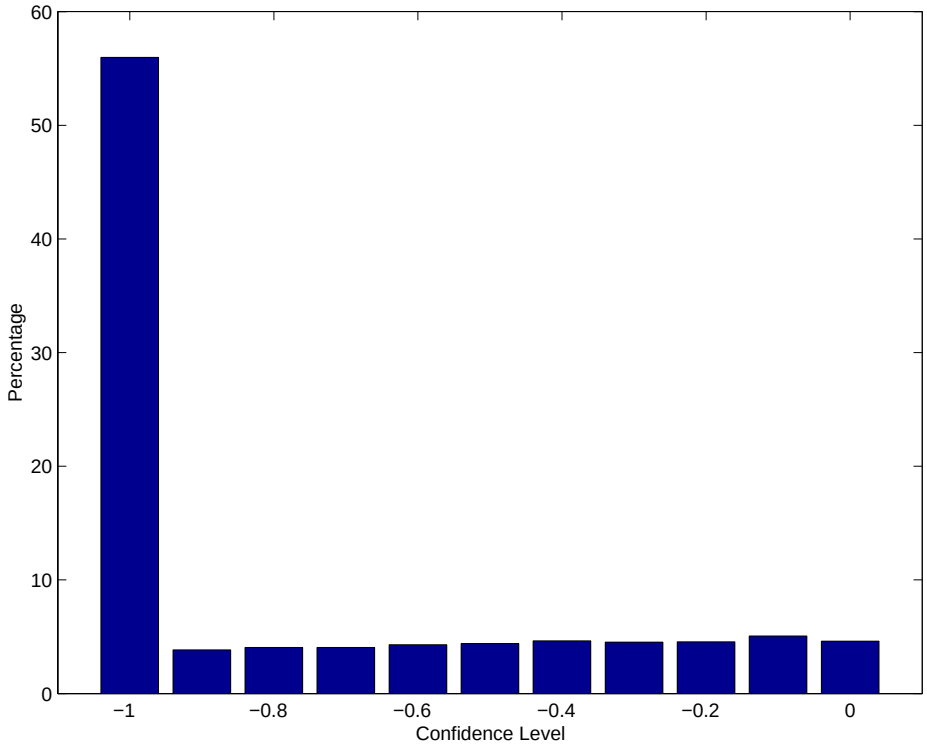

Figure 6.4: Efficiency of confidence measures for correctly rejected speakers.

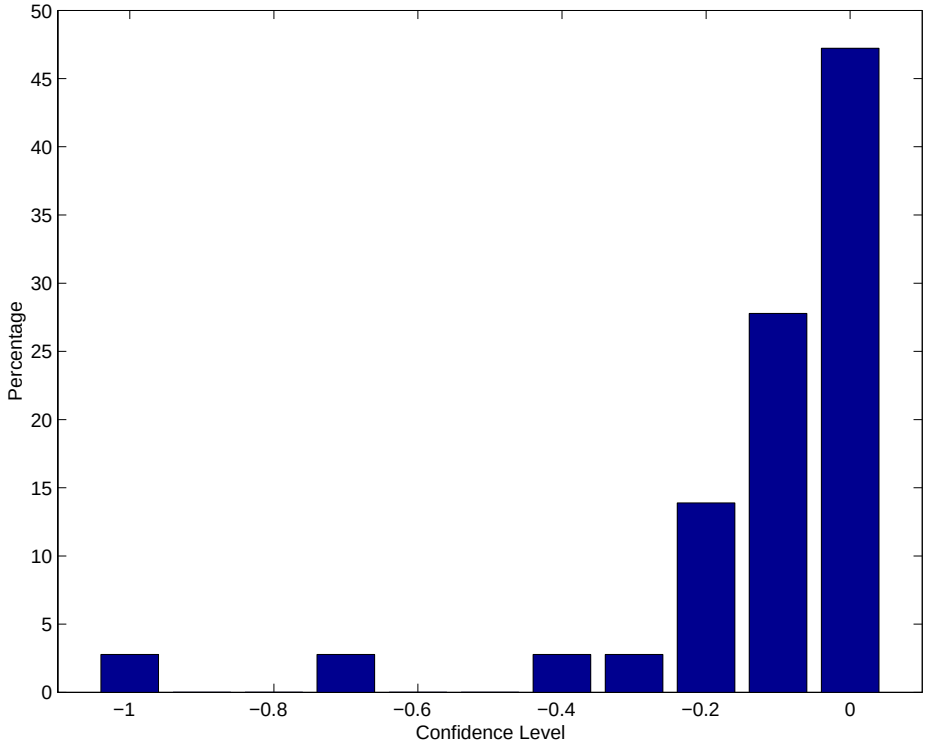

Figure 6.5: Efficiency of confidence measures for false rejections.

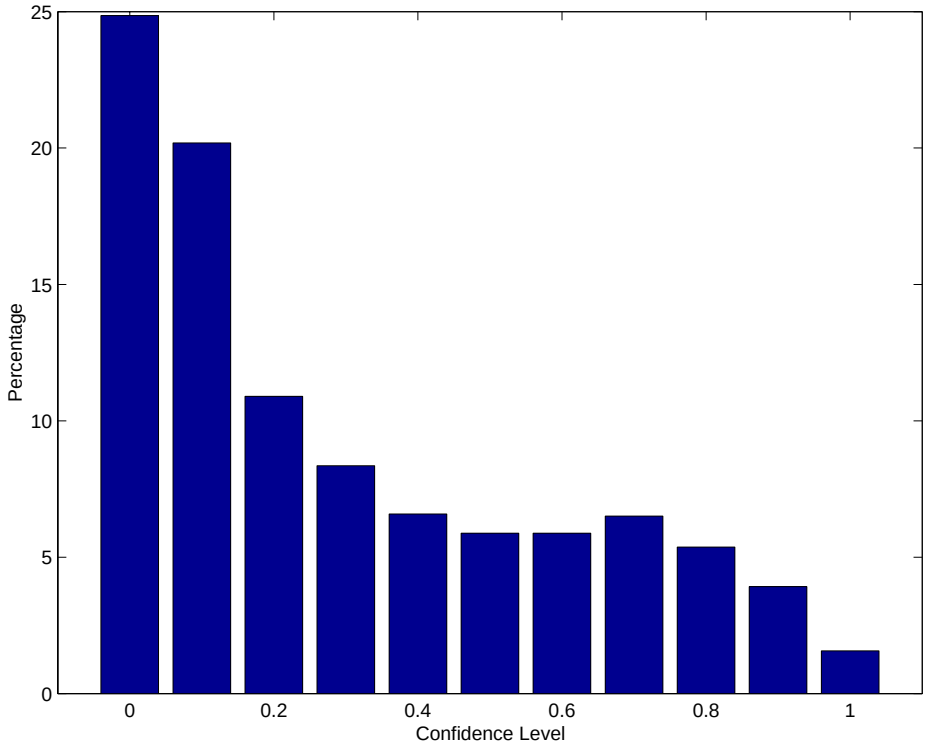

Figure 6.6: Efficiency of confidence measures for false acceptances.

Chapter 7

Use of Confidence Measures for Speaker Model Adaptation in Speaker Verification

As stated in previous chapter, adaptation is an important step of speaker verification process which is, unlike training step, used in entire life cycle of a speaker verification system. The need for adaptation comes from changes in acoustic conditions of verification environment and in the voice of some of the speakers.

The purpose of speaker model adaptation is to improve the accuracy of speaker verification systems when there are variabilities occurred after the end of training phase. Those variabilities will cause performance reduction of the system.

Adaptation of a statistical model means re-estimation of model parameter to cover the adaptation data. In text independent speaker verification generally GMM based speaker models are used. The parameters of GMMs, means and standard deviations, are updated with new data during adaptation procedure.

State-of-the-art adaptation techniques are based on transformation of the speaker models. MAP [79] and MLLR [80] are most popular model transformation methods used for adaptation of GMM based speaker models. A detailed comparison of MAP and MLLR techniques will be given later in this chapter.

Speaker model adaptation can be applied in two ways:

1. Supervised adaptation is applied when the speaker of adaptation data is known. This type of adaptation can be considered as a further step for the training phase.

2. Unsupervised adaptation is applied when the source of adaptation data is unknown and the classification of adaptation data is obtained from the speaker verification system.

The second type of adaptation is more useful than the first one since it allows online adaptation of speaker models without any intervention. This property of Unsupervised adaptation

make it harmful when it is not carefully applied. Confidence measures are used to minimize the risk of adapting speaker models to the adaptation data coming from an impostor.

7.1 GMM Based Acoustic Speaker Modeling

GMM based speaker modeling is a good performing and efficient technique for text independent speaker recognition systems [42]. Use of GMM technique for speaker recognition is explained in chapter 3. In this chapter we will give some details about EM algorithm which is used for training speaker models based on GMM.

GMM is a parametric statistical model, the parameters need to be estimated from sample data during training phase. A speaker GMM is defined as three tuple model as:

$$\lambda = \{p_i, \mu_i, \Sigma_i\}, i = 1, ..., M \tag{7.1}$$

where λ is the speaker model, M is the number of components in the mixture, p_i is the weight coefficient of the component i, b_i is probability density for the component i computed as explained in chapter 3, μ_i and σ_i are the mean and standard deviation for the the component i. It must be noted that p_i is the probability value assigned to component i and $\sum_i p_i = 1$. Σ_i is the covariance matrix which include correlation between M Gaussians. Covariance matrix is usually selected as diagonal meaning that Gaussians are independent from each other.

The estimation of GMM parameters is based on maximum likelihood principle which use EM algorithm. The parameters are estimated for maximizing the probability:

$$p(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda) \tag{7.2}$$

where $p(X|\lambda)$ is the probability that speaker model λ matches the data X, T is the number of feature vectors and x_t is the feature vector with index t. This probability is usually computed in logarithmic domain as:

$$\log\left(p(X|\lambda)\right) = \sum_{t=1}^{T} \log\left(p(x_t|\lambda)\right) \tag{7.3}$$

and is the maximum likelihood for the data X given the speaker model λ. The maximization is performed by EM algorithm.

The two important steps of EM algorithm are initialization and re-estimation steps. In initialization step, acoustic vectors are clustered by well-known k-means technique [82]. The means, variances and weights of Gaussians are initialized from the position of the centroïds and the covariance matrices of each cluster.

After the initialization, re-estimation is realized by following formulas:

$$\overline{p}_i = \frac{1}{T} \sum_{t=1}^{T} p(i|x_t, \lambda) \tag{7.4}$$

$$\overline{\mu}_i = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda) x_t}{\sum_{t=1}^{T} p(i|x_t, \lambda)} \tag{7.5}$$

$$\overline{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda) x_t^2}{\sum_{t=1}^{T} p(i|x_t, \lambda)} - \overline{\mu}_i^2 \tag{7.6}$$

where σ_i, x_t, and μ_i represent the the standard deviation vector $\vec{\sigma_i}$, the feature vector $\vec{x_t}$, and the mean vector $\vec{\mu_i}$ respectively. The probability $p(i|x_t, \sigma)$ is defined as:

$$p(i|x_t, \sigma) = \frac{p_i b_i(x_t)}{\sum_{k=1}^{M} p_k b_k(x_t)} \tag{7.7}$$

where p_i is the weight and $b_i(x_t)$ is the probability density computed from the pdf of the Gaussian. Those values are computed by using the parameters of the initial model. The re-estimation procedure is an iterative procedure, initial model is replaced by newly estimated model and the iteration is stopped when there is no more increase in the maximum likelihood.

The critical factors in training a GMM based speaker model are:

- Initialization. A good initialization prevent finding local maximum likelihoods during EM algorithm.
- Selection of a sufficiency large component size (M). When more data is available, the number of mixture components must be higher. This will result in better representation of feature space.

An important property of GMM based speaker modeling is that when there is more training data available and the training data represents well enough the speaker characteristics, the resulting model can attains high classification performances. But the performance can drop rapidly when there are mismatches between training and testing conditions.

7.2 MAP and MLLR Adaptation

This section give some details about most popular speaker model adaptation techniques: MAP and MLLR. The experiments and performance evaluations of confidence measure based unsupervised adaptation in this chapter are based on these two adaptation techniques.

The performance of MAP adaptation will be increased when more adaptation data are available, but when the amount of adaptation data is low, the adapted model will be a

specialized version of initial model which will cause performance reductions for "unseen" data.

MLLR adaptation, unlike MAP, can be used even if the available adaptation data is small. Transformation matrix update only the model parameters related to "seen" data.

7.2.1 MAP

Speaker model adaptation with MAP technique is based on use of prior knowledge about the model and the adaptation data to obtain an adapted model. The speaker model obtained in training phase is used as a base for the new model and the parameters of prior distributions are updated to obtain a more general model that covers the newly observed data. The use of prior model and a predetermined weighing factor prevents over fitting of the speaker model to newly observed data which cause degradation in overall accuracy of the system.

In MAP adaptation, the a posteriori probability that the model λ matches the observation O is maximized by updating the parameters of initial model [83]. The adapted model is defined as:

$$\lambda_{MAP} = arg \max_{\lambda} f(\lambda|O) \tag{7.8}$$

The equation (7.8) is replaced by its equivalent obtained by applying Bayes' rule. The new adaptation formula is:

$$\lambda_{MAP} = arg \max_{\lambda} \frac{L(O|\lambda)P_0(\lambda)}{P(O)} \tag{7.9}$$

where $L(O|\lambda)$ is the likelihood probability of the observation sequence O given the speaker model λ, $P(O)$ is the prior probability of observing O and is ignored since it remains constant for training and adaptation data. $P_0(\lambda)$ is the probability density obtained from the pdf of initial model.

MAP adaption can be either used for variance adaptation or mean adaptation. The adaptation of Gaussian means of the GMM is the more efficient than adaptation of variances [15]. The update formula for adaptation of means is defined as:

$$\hat{\mu}_m = \frac{N_m}{N_m + \tau}\overline{\mu}_m + \frac{\tau}{N_m + \tau}\mu_m \tag{7.10}$$

where τ is the weighing factor used for prior model, m is the index of mixture component, μ_m is the mean parameter for the initial model, $\overline{\mu}_m$ is the mean value computed for the adaptation data and $\hat{\mu}_m$ is the mean of adapted model. N_m is the occupation likelihood of the adaptation data defined as:

$$N_m = \sum_{t=1}^{T} L_m(t) \tag{7.11}$$

where $L_m(t)$ is the likelihood value for the frame t of the observation sequence composed of T frames.

The mean value computed for adaptation data, $\overline{\mu}_m$, in equation (7.10) is computed as:

$$\overline{\mu}_m = \frac{\sum_{t=1}^{T} L_m(t) o_t}{\sum_{t=1}^{T} L_m(t)} \tag{7.12}$$

where o_t is an the observation vector for the frame t.

Note that the equations (7.11) and (7.12) are applied for one state GMM based speaker models. It is possible to use more than one state for GMM based text independent speaker verification, for example silence/speech states, voiced/unvoiced states, ... When there are more than one state, the sums on the equations (7.11) and (7.12) are taken for all available states and the sum of all the sums is used.

The equation (7.12) shows that when the likelihood of observation is higher, the mean for the observation value will be high, which will result in more important adaptations because of the increasing effect of this mean in equation (7.10). The τ in the equation (7.10) can be used to prevent rapid changes in the mean values of GMMs when the amount of adaptation data is low.

7.2.2 MLLR

MLLR adaptation of speaker models consists in producing a set of regression based transformations based on adaptation data and use these transformations to update GMM parameters. As it was the case for MAP adaptation, MLLR adaptation also is used generally only for adapting the means of GMMs which are considered as most important components of GMMs for adaptation.

The MLLR adaptation is based on computing a transformation matrix and use it to update the means of the model. The new GMM means are obtained as follows:

$$\hat{\mu}_s = W_s \xi_s \tag{7.13}$$

where W_s is an $n \times (n+1)$ transformation matrix (n is the dimension of feature vector) and ξ_s is the extended mean vector. ξ_s is defined as:

$$\xi_s = [w, \mu_{s_1}, \mu_{s_2}, ..., \mu_{s_n}]^T \tag{7.14}$$

where w is the offset term of the regression ($w = 1$ means include offset, $w = 0$ means ignore offsets) and $[..]^T$ indicates matrix transpose operation.

W_s is computed by solving the following equation:

$$\sum_{t=1}^{T} \sum_{r=1}^{R} L_{s_r}(t) \Sigma_{s_r}^{-1} o(t) \xi'_{s_r} = \sum_{t=1}^{T} \sum_{r=1}^{R} L_{s_r}(t) \Sigma_{s_r}^{-1} W_s \xi_{s_r} \xi'_{s_r} \tag{7.15}$$

where T is the number of frames in the observation data, R is the number of states. The implementation issues for MLLR can be seen in [80].

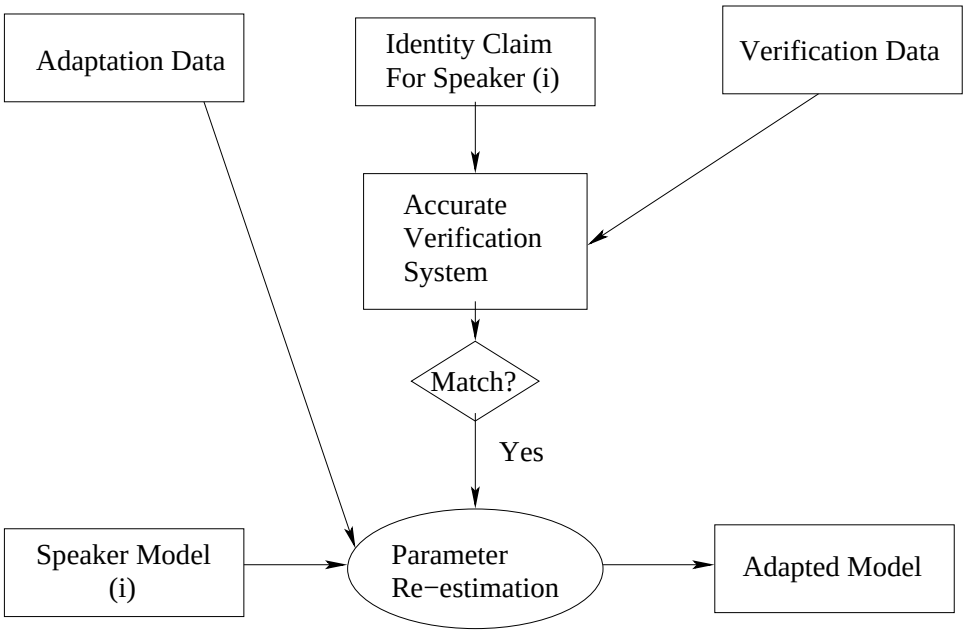

Figure 7.1: Supervised speaker model adaptation

7.3 Unsupervised Adaptation with Confidence Measures

Unsupervised adaptation of speaker models is used when the true source of observation data is unknown by the speaker verification system. The alternative for this type of adaptation could be supervised adaption in which the identity of the speaker could be obtained by an accurate identification system like finger print verification, iris structure verification, access card verification or by a human agent.

Figure 7.1 shows a supervised speaker model adaptation. The parameter re-estimation procedure can be one of the adaptation method explained above.

In unsupervised adaptation there is no separate verification data, the adaptation data is used for verification purpose also. The correctness of claimed speaker identity, is determined by speaker verification system. A simple unsupervised adaptation can be adapting speaker model each time an access decision is taken by the speaker verification system. This adaptation method can be dangerous for the system because when there is a "false acceptance", the speaker model will be adapted to the data coming from an impostor. This adaptation will cause a performance reduction of the system.

To avoid the problem of adaptation with erroneous data a fuzzy criterion can be used to select the adaptation data:

. "Adapt the model when it is highly probable that utterance comes from the correct speaker".

This fuzzy statement can be verified by applying confidence measures on the decision of the

speaker verification systems [81]. In other words, the confidence level of the decision taken by the speaker verification system is computed and if this confidence level is higher than a confidence limit, the utterance is judged "it is highly probable that the utterance comes from true speaker".

7.4 Experiments

Experiments are realized to compare two adaptation methods and the effect of confidence measures for adaptation of speaker models in text independent speaker verification. The POLYCOST [84] speaker recognition database is used for experiments.

Three method for GMM training, as explained in the previous chapter, are used:

1. One state GMMs for each speaker and for the world model,
2. Two state GMMs, one for speech and one for silence parts of the data. Silence part is ignored in likelihood computations,
3. two state GMMs, one for voiced and one for unvoiced parts of the data. Unvoiced part is ignored in likelihood computations.

Two types of adaptation data are used in the experiments, clean adaptation data and noisy adaptation data. Noisy adaptation data is obtained by adding Gaussian white noise (SNR=15) to the clean adaptation data. The results for male and female speakers are listed separately because the different adaptation methods have different effects on male and female speaker models.

7.5 Results and Discussion

Results obtained from the experiments described above are shown in six different group of bar graphics. Each group includes the results for female and male speakers. Error rates for three modeling types, 1-state GMMs, speech/silence GMMs and voiced/unvoiced GMMs, are shown in the bar graphics. MLLR adaptation and MAP adaptation is compared for certain groups.

Figure 7.2 and Figure 7.3 shows the error rates for different modeling types used for speaker models. The test data set used to obtain these results have the same characteristics as the training data set, the error rates obtained in those two figures are the minimum error rates obtained by the speaker verification system used in this thesis.

Figure 7.4 and Figure 7.5 shows the error rates after adapting the models to some clean speech adaptation data. The figures show that the error rates obtained for adapted models is higher than the initial models. This situation can be explained by use of insufficient adaptation data. The amount of adaptation data is selected as low in order to test the

efficiency of model adaptations with few data. It must be noted that the performance of MLLR adaptation is better than MAP adaptation because MAP adaptation need more data for higher performances. The remaining part of this section reports only adaptation results for MLLR adaptation. The performances of male speaker models is higher than the female models for all the tests realized on POLYCOST database.

Figure 7.6 and Figure 7.7 shows the error rates for noisy speech with the speaker models trained on clean speech. As it can be seen from figures, the increase on female error rates is more important the male error rates. This can be explained by vulnerability of speech data obtained from female speakers to noise. It can be seen also from the figures that the error rates for two state GMM methods are higher because presence of noise in the test data cause some difficulties in classification of frames as speech/silence or voiced/unvoiced.

Figure 7.6 and Figure 7.7 shows the error rates for speaker models trained on noisy data. The performances shown on these figures can be considered as target performances for adaptation methods.

Figure 7.10 and Figure 7.11 shows the error rates for noise adapted speaker models. The adaptation method used is MLLR adaptation. All the available adaptation data is used for adaptation. As it can be seen from the figures, error rates are decreased when they are compared to figures 7.6 and 7.7. Reduction in error rates show that the adaptation is efficient. Reduction in male speaker error rates is more important than reduction in female speaker error rates, this can be explained by loss of some parts of speech data from female speakers occurred by additive noise. We can see that two state GMMs are still less performant than 1-state GMMs.

Figure 7.12 and Figure 7.13 shows the effect of confidence measure usage for adaptation. The adaptation data is used when the confidence level threshold, 50%, is obtained. It can be seen from the figures that the target performances of figures 7.8 and 7.9 are reached and higher performances are obtained for female and male speakers. The performances obtained by confidence measure based adaptation are close to, even better for female speakers, the supervised adaptation performances of figures 7.10 and 7.11.

The results show that use of confidence measure for speaker adaptation improve the performances of speaker models in noisy conditions. Use of two state GMM based speaker modeling, speech/silence or voiced/unvoiced, provide good performances only when the training and testing conditions are free of noise. It can be seen also that MAP adaptation does not improve the performances of speaker models when there are few adaptation data available.

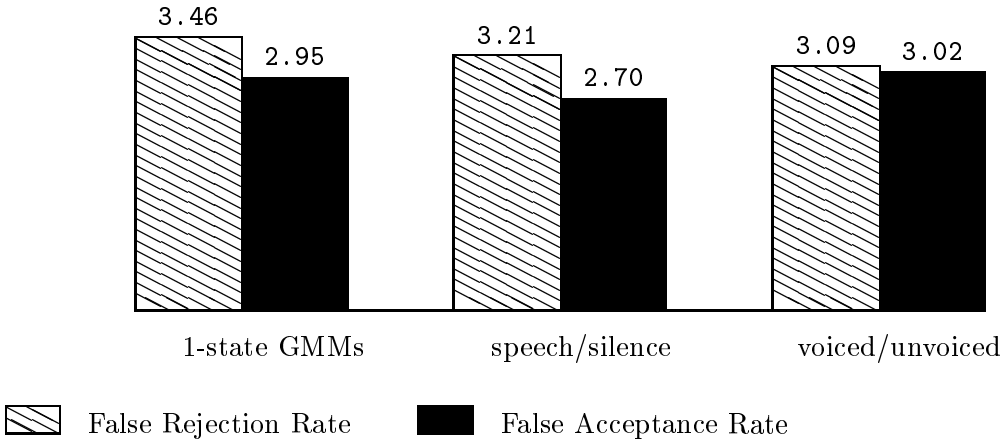

Figure 7.2: Clean speech error rates for female speaker models

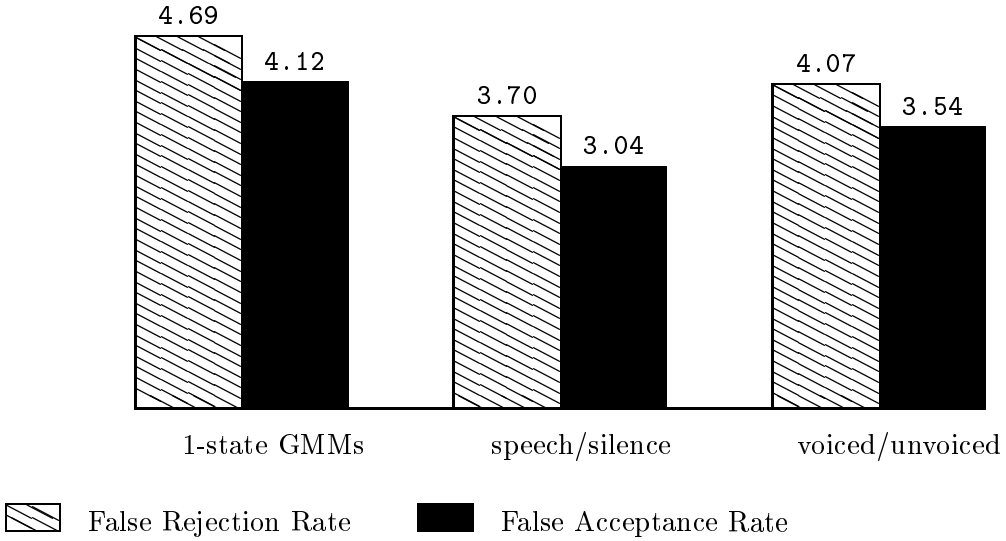

Figure 7.3: Clean speech error rates for male speaker models

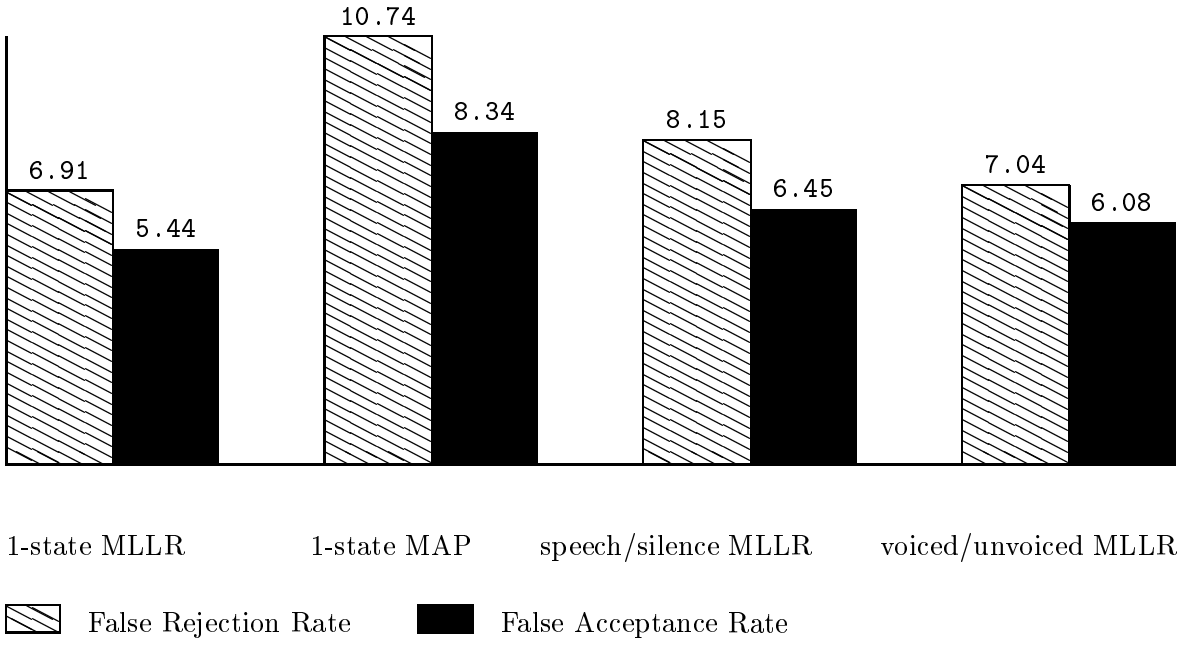

Figure 7.4: Error rates for female speaker models after adaptation with clean speech

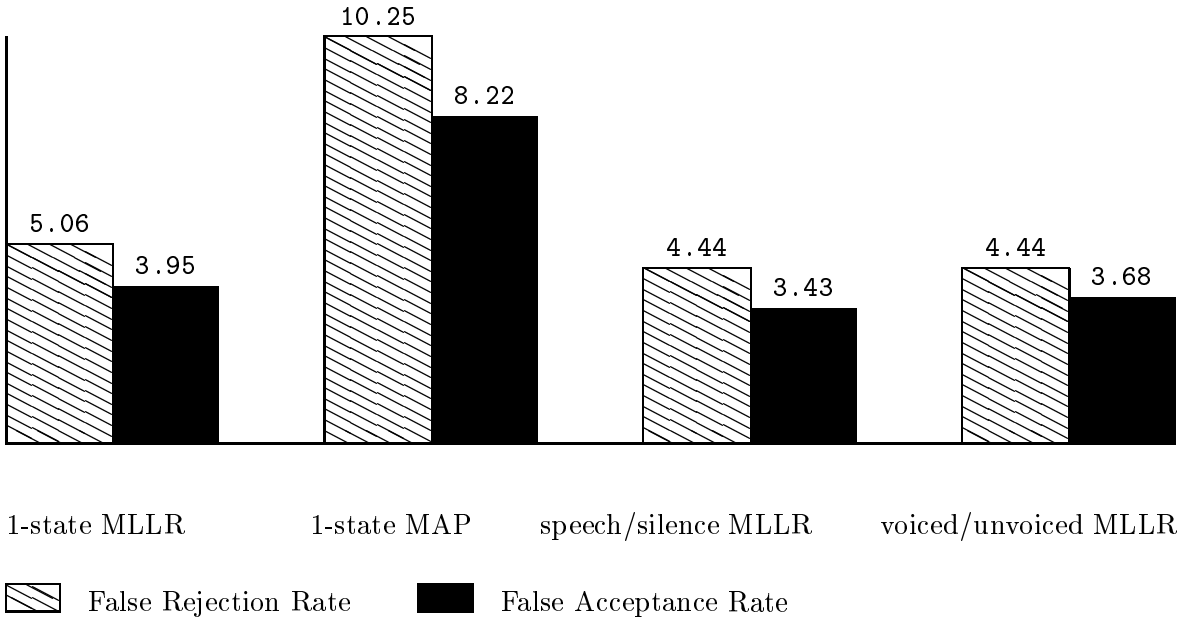

Figure 7.5: Error rates mor male speaker models after adaptation with clean speech

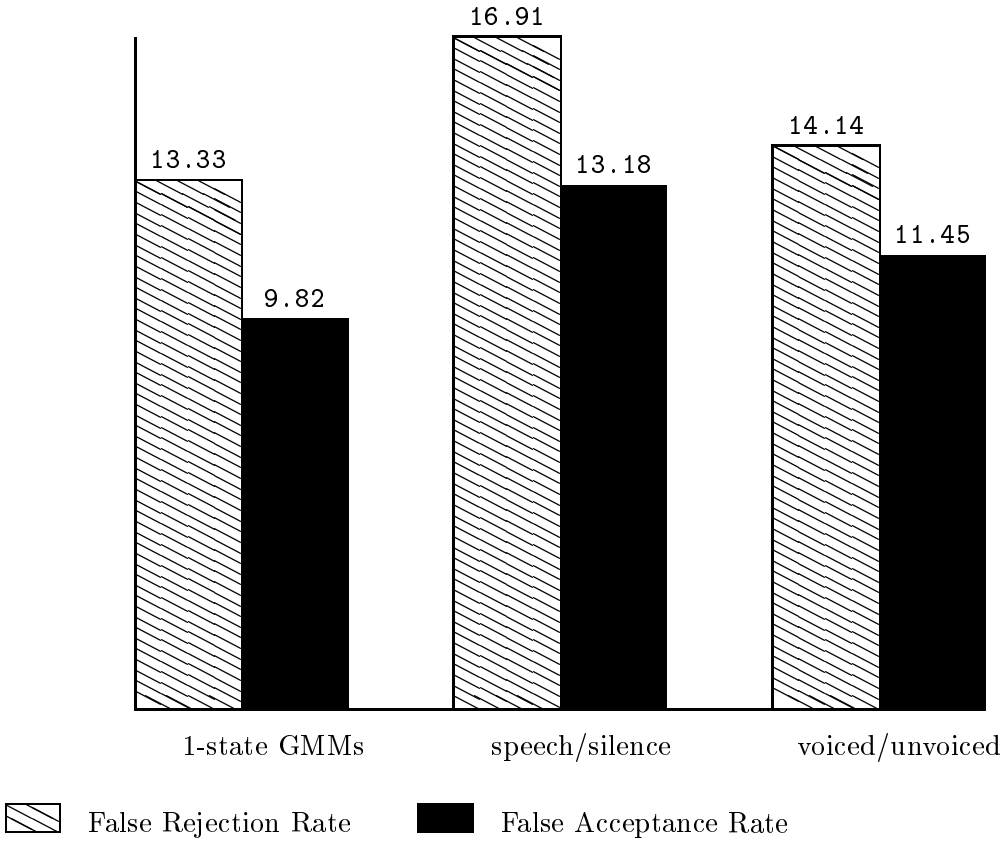

Figure 7.6: Noisy speech error rates for female speaker models

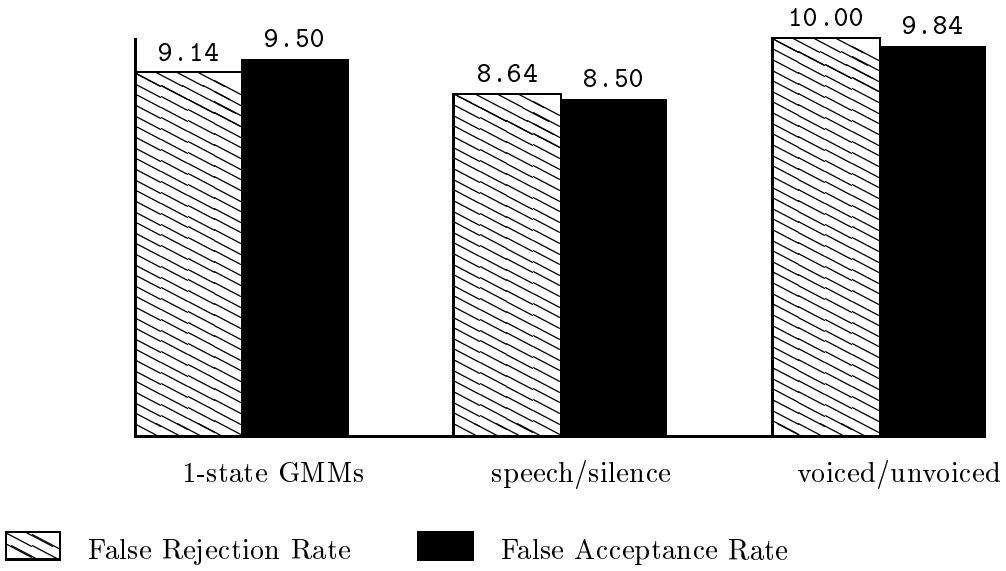

Figure 7.7: Noisy speech error rates for male speaker models

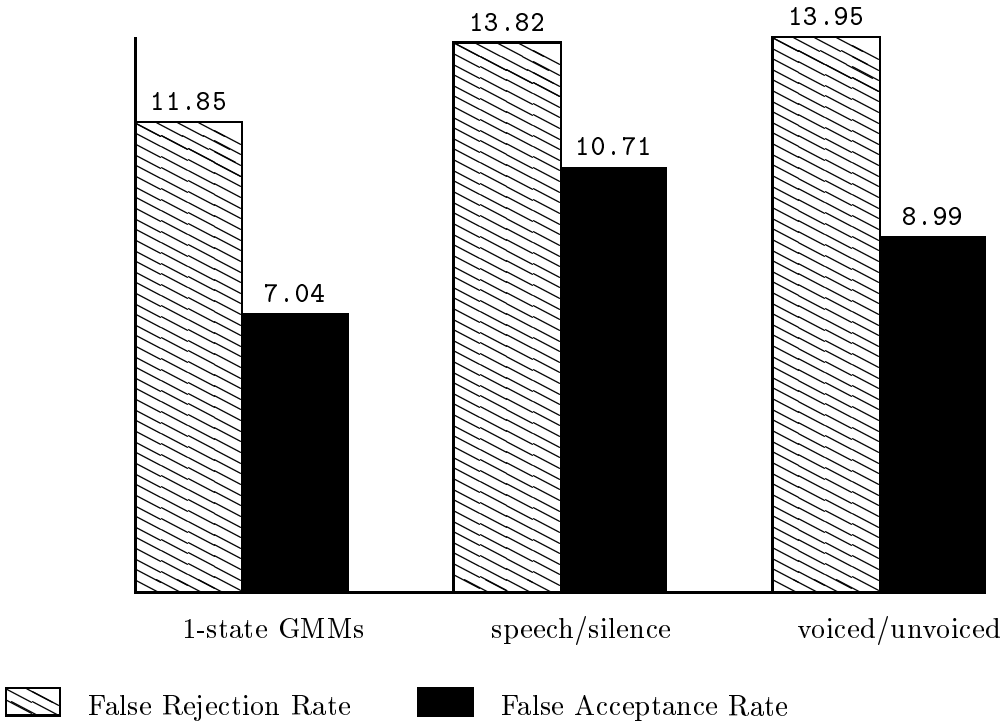

Figure 7.8: Noisy speech error rates for female speaker models trained on noisy data

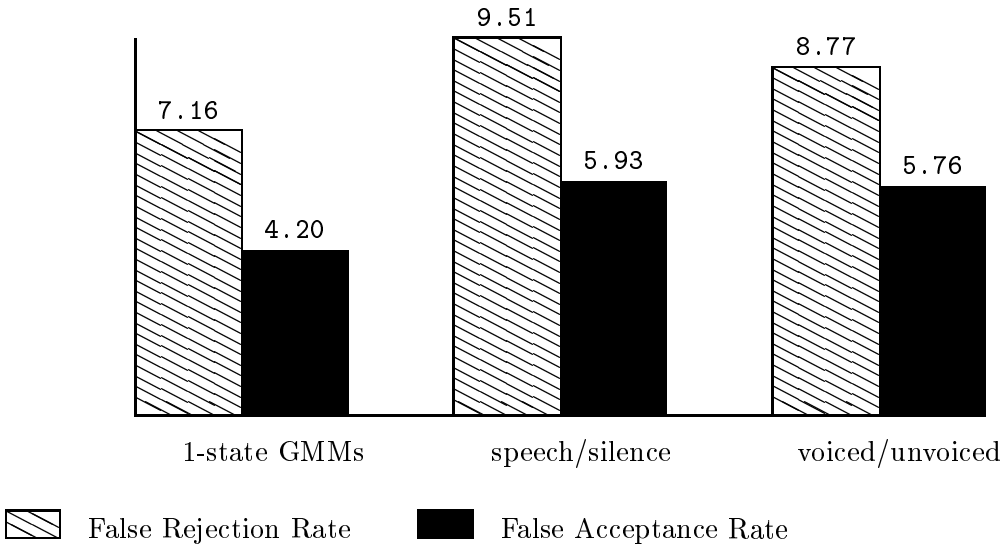

Figure 7.9: Noisy speech error rates for male speaker models trained on noisy data

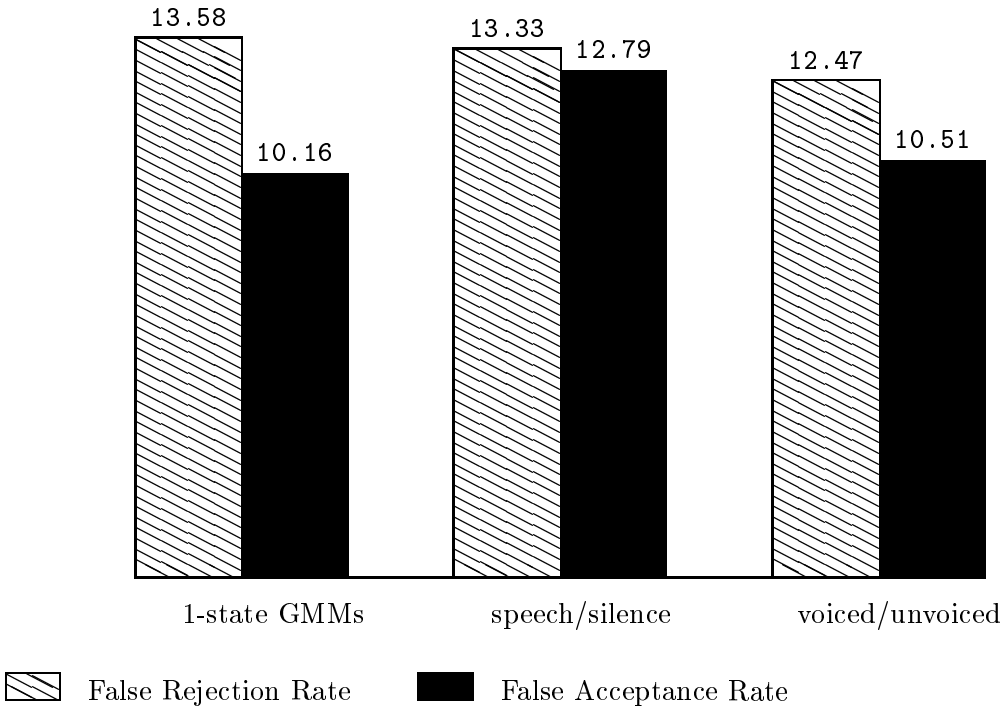

Figure 7.10: Noisy speech error rates for noise adapted female speaker models (without confidence measures)

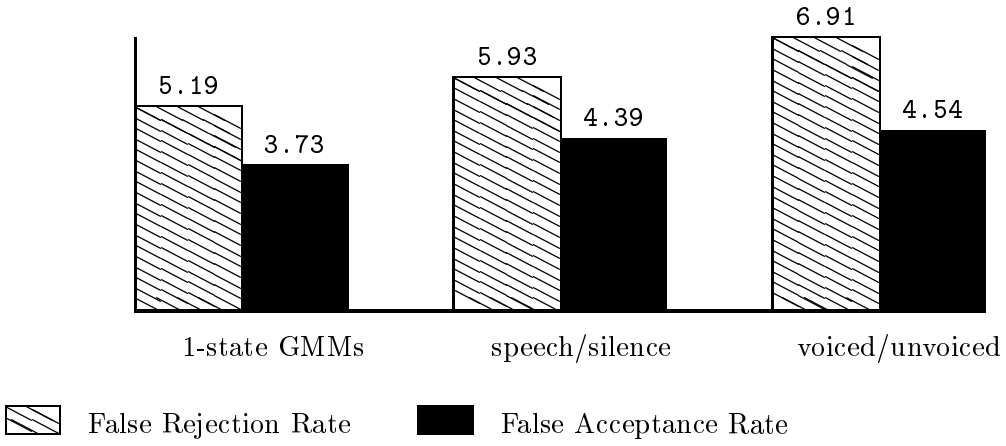

Figure 7.11: Noisy speech error rates for noise adapted male speaker models (without confidence measures)

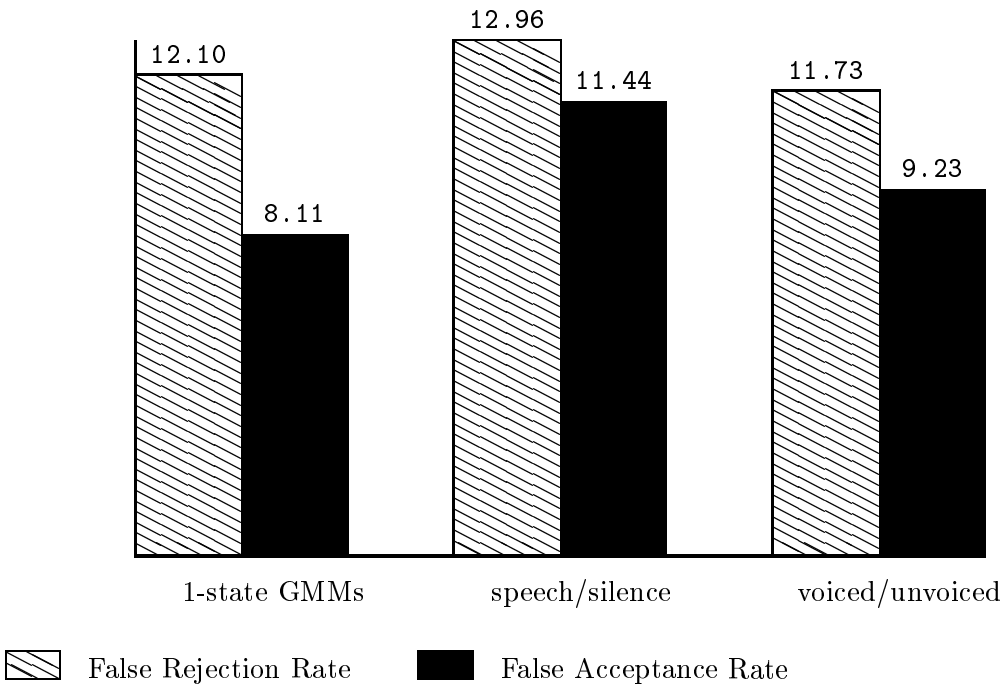

Figure 7.12: Noisy speech error rates for noise adapted female speaker models (with confidence measures)

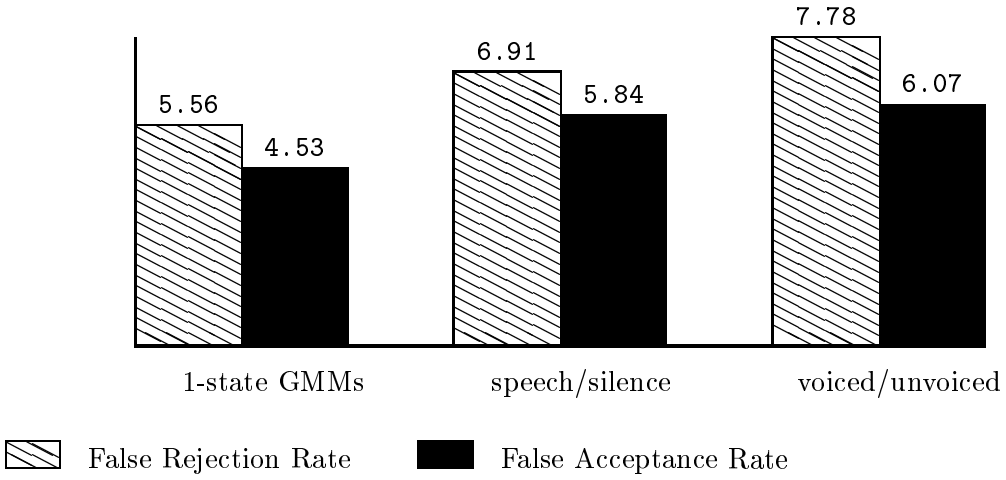

Figure 7.13: Noisy speech error rates for noise adapted male speaker models (with confidence measures)

Chapter 8

Conclusions

This thesis provides a new language modeling technique for the Turkish language, which is an agglutinative language, and new confidence measures for speech/speaker recognition systems. The efficiency of the newly proposed language model and newly defined confidence measures is compared to the methods in the literature.

A Turkish continuous speech recognition database is also created and used for developing an HMM/MLP hybrid Turkish speech recognition system.

The language modeling technique proposed in this thesis for agglutinative languages use the high inflection property of the Turkish language to decompose the words into stems and endings. The resulting decomposition is used for n-gram modeling of the language. The results obtained in experiments showed that the perplexity of the language model decreases substantially with the new technique and that the rate of out-of-vocabulary words is very low when compared to the original n-gram language modeling which uses the entire words. The entropy of the new language model is observed as lower than the classical n-gram modeling.

We also test also the efficiency of a hybrid HMM/MLP speech recognition system for the Turkish language which is one of the least studied language for speech recognition. We determine baselines for unconstrained large vocabulary continuous speech recognition and isolated word recognition tasks.

The other contribution of this thesis is the newly defined confidence measure for speech recognition. Acoustic prior information based confidence measure is tested on an isolated word speech recognition task and compared to other confidence measure techniques in the literature. The results and the discussions are included in the corresponding chapter.

The efficiency of confidence measures in speaker recognition is also tested. Confidence measures are used for selecting adaptation data for speaker model adaptation. The results show that use of confidence measures improve the efficiency of adaptation. A new interpretation method based on Fisher transformation is introduced for confidence measures on speaker verification task. A new interpretation method provide linear interpretation of confidence measures. Further researches are needed to evaluate the efficiency of this new interpretation

for dialogue management or other speech recognition problems.

In this thesis we investigated acoustic model based confidence measures. Combination of acoustic model based confidence measures and language model based confidence measures remains to be investigated. An efficient combination of confidence measures from these two models should use the n-best hypothesis list provided by the speech recognizer. Combined confidence measure can be used as a weighing factor for selection of next word given the history.

Bibliography

[1] Sheryl R. Young, "Detecting Misrecognitions and Out-Of-Vocabulary Words", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1994.

[2] Sheryl R. Young, "Recognition Confidence Measures: Detection of Misrecognitions and Out of Vocabulary Words", *Technical Report*, Carnegie Mellon University, School of Computer Science, CMU-CS-94-157, 1994.

[3] Charles W. Anderson, Micheal J. Kirby, "EEG Subspace Representations and Feature Selection for Brain-Computer Interfaces", *Proceedings of the 1st IEEE Workshop on Computer Vision and Pattern Recognition for Human Computer Interaction (CVPRHCI)*, Medison, Wisconsin, June 17, 2003.

[4] Hynek Hermansky, "Should Recognizers Have Ears", *Speech Communication*, 25:3-27, 1998.

[5] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 1989.

[6] René Boite, Murat Kunt, *Traitement de la parole*, Press Polytechnique Romandes, 1987.

[7] Steven B. Davis, Paul Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", *IEEE Transactions On Acoustics, Speech, and Signal Processing*, 28:357-366, 1980.

[8] Hynek Hermansky, Nelson Morgan, "RASTA Processing of Speech", *IEEE Transactions On Speech and Audio Processing 2(4):578-589*, I994.

[9] A.K. Jain, R.P.W. Duin, Jianchang Mao, "Statistical pattern recognition: a review" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4-37, 2000.

[10] S. Dupont, H. Bourlard, O. Deroo, V. Fontaine, and J. M. Boite, (1997), "Hybrid HMM/ANN Systems for Training Independent Tasks: Experiments on 'Phonebook' and Related Improvements,", *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (Munich, Germany)*, pp. 1767-1770, 1997.

[11] K. Paliwal, B. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", *IEEE Transactions on Speech and Audio Processing*, 1(1):3-14, 1993.

[12] R. M. Gray, "Vector Quantization" *IEEE ASSP Magazine*, 1:4-29, April 1984.

[13] Erhan Mengusoglu, "Rule Based Design and Implementation of a Speech Recognition System for Turkish Language", *Master of Science Thesis*, Hacettepe University, Institute of Applied Sciences, 1999.

[14] Erhan Mengusoglu, Harun Artuner, "Using Multiple Codebooks for Turkish Phone Recognition", 14th International Symposium on Computer and Information Sciences, Izmir, Turkey, 1999.

[15] Stece Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Valtcho Valtchev, Phil Woodland, *The HTK Book*, Cambridge University Engineering Department, 2002.

[16] Frederick Jelinek, *Statistical Methods for Speech Recognition*, The MIT Press, England, 1998.

[17] Kurt Hornik, Maxwell Stinchcombe, Halbert White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, 2(5):359-366, 1989.

[18] H. Bourlard, N. Morgan, *Connectionist Speech Recognition*, Kluwer Academic Pu2.132.417 blishers, London, 1994.

[19] John-Paul Hosom, Ron Cole, Mark Fanty, Johan Schalkwyk, Yonghong Yan, Wei Wei, "Training Neural Networks for Speech Recognition" *Technical Report*, Center for Spoken Language Understanding (CSLU) Oregon Graduate Institute of Science and Technology, 1999.

[20] Les Niles, Harvey Silverman, Gary Tajchman, Marcia Bush, "How limited training data can allow a neural network to outperform an optimal statistical classifier", *Proc. ICASSP'89*, pp. 17 - 20, May 1989.

[21] R. P. Lippmann, "Review of neural networks for speech recognition", *Neural Computation*, 1:1-38, 1989.

[22] James A. Freeman, David M. Skapura, *Neural networks : algorithms, applications, and programming techniques* Addison-Wesley, 1991.

[23] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, *Spoken Language Processing*, Prentice Hall, New Jersey, 2001.

[24] Slava M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", *IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-35(3):400-401*, March 1987.

[25] Adwait Ratnaparkhi, *Maximum Entropy Models for Natural Language Ambiguity Resolution*, Ph.D. Dissertation, University of Pennsylvania, 1998.

[26] Doug Cutting, Julian Kupiec, Jan Pedersen, Penelope Sibun, "A Practical part-of-speech tagger", *Proceedings of the Third Conference on Applied Natural Language Processing*, 1992.

[27] Thorsten Brants, "TnT - A statistical Part-Of-Speech Tagger", *Proceedings of the Sixth Applied Natural Language Processing Conference, ANLP-2000*, 2000.

[28] Peter A. Heeman, "POS tags and decision trees for language modeling", *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, College Park, Maryland*, June 1999.

[29] S. Deketelaere, T. Dutoit, O. Deroo, "Speech Processing for Communications : what's new?", *Revue HF*, March 2001.

[30] Brown, M.K. McGee, M.A. Rabiner, L.R. Wilpon, J.G., "Training set design for connected speech recognition" *IEEE Transactions on Signal Processing*, 39(6):1268-1281, 1991.

[31] Mukund Padmanabhan, Michael Picheny, "Large-Vocabulary Speech Recognition Algorithms", *IEEE Computer*, 35(4):42-50, 2002.

[32] Knill, K.M., and S.J. Young, "Speaker Dependent Keyword Spotting for Accessing Stored Speech, *Technical Report CUED/F-INFENG/TR 193*, Cambridge, U.K.: Cambridge University, 1994.

[33] A. Martin, T. K. G. Doddington, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance", *Proceedings of EuroSpeech '97*, 4:1895-1898, 1997.

[34] Douglas A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models", *Speech Communication*, 17(1-2):91-108, August 1995.

[35] Lan Wang, Ke Chen, and Huisheng Chi, "Capture Interspeaker Information With a Neural Network for Speaker Identification", *IEEE Transactions on Neural Networks*, 13(2):436-445, March 2002.

[36] Toshihiro Isobe, Jun-ichi Takahashi, "A New Cohort Normalization Using Local Acoustic Information For Speaker Verification" *ICASSP*, 1999.

[37] Johan Lindberg and Hakan Melin, "Text-prompted versus sound-prompted passwords in speaker verification systems", *EUROSPEECH*, 1997

[38] Sadaoki Furui, "Cepstral Analysis Technique for Automatic Speaker Verification" *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-29(2):254-272, APRIL 1981.

[39] Joseph A. Campbell, "Speaker Recognition: A Tutorial", *Proceedings of the IEEE*, 85(9):1437-1462, September 1997.

[40] Erhan Mengusoglu, Henri Leich, "Reconnaissance de la Parole / du Locuteur dans le Domaine Medical", (Book Chapter) *ARTHUR, Manuel d'informatisation des urgences hospitalières*, edited by Jean Herveg and Anne Rousseau, UCL Press, October 2003.

[41] Jeff A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", *Technical Report*, International Computer Science Institute Berkeley CA, April 1998.

[42] Douglas A. Reynolds, Richard C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE Transactions on Speech and Audio Processing*, 3(1):72-83, January 1995.

[43] Bojan Nedic, Herve Bourlard, "Recent Developments in Speaker Verification at IDIAP", *Technical Report*, IDIAP, Spetember 2000.

[44] Sadaoki Furui, "Recent advances in speaker recognition", *Pattern Recognition Letters*, 18:859-872, 1997.

[45] Kwok-Kwong Yiu, Man-Wai Mak, Sun-Yuan Kung, "Environment Adaptation for Robust Speaker Verification", *EUROSPEECH 2003 - Geneva*, 2003.

[46] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, 1986.

[47] R. Kuhn, P. Nguyen, J.-C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, M. Contolini, "Eignevoices for Speaker Adaptation", *ICSLP*, 1998.

[48] Dilek Z. Hakkani-Tür, Kemal Oflazer , Gökhan Tür, "Statistical Morphological Disambiguation for Agglutinative Languages", *Technical Report*, Bilkent University, Computer Engineering, BU-CE-0002, January 2000.

[49] Kemal Oflazer, "Two-level Description of Turkish Morphology", *Literary and Linguistic Computing*, 9(2):137-148, 1994.

[50] I. Maddieson, *Patterns of Sounds*, Cambridge University Press, 1984.

[51] Department of Phonetics and Linguistics, University College London, *http://www.phon.ucl.ac.uk/home/sampa/turkish.htm*, UK, 2002.

[52] Cemal Yilmaz, "A Large Vocabulary Speech Recognition System for Turkish", Bilkent University, 1999.

[53] Beryl Hoffman, "Word Order, Information Structure, and Centering in Turkish", *Centering Theory in Discourse*, Oxford University Press, UK, 1997.

[54] Johan Carlberger, Viggo Kann, "Implementing an efficient part-of-speech-tagger", *Software Practice and Experience*, 29:815-832, 1999.

[55] W. Byrne, J. Hajic, P. Ircing, P. Krbec, J. Psutka, "Morpheme Based Language Models for Speech Recognition of Czech", *TSD 2000 - Third International Workshop on TEXT, SPEECH and DIALOGUE*, Brno, Czech Republic, September 13-16, 2000.

[56] Dimitri Kanevski, Salim Estephan Roukos, *Statistical Language Model for Inflected Languages*, US Patent No:5,835,888 1998.

[57] Erhan Mengusoglu, Olivier Deroo, "Turkish LVCSR: Database Preparation and Language Modeling for an Agglutinative Language", *ICASSP 2001*, Student Forum, 2001.

[58] Erhan Mengusoglu, Olivier Deroo, "Confidence Measures in HMM/MLP Hybrid Speech Recognition for Turkish Language", *PRORISC*, 2000.

[59] Alberto Leon-Garcia, *Probability and Random Process for Electrical Engineering*, Addison-Wesley Publishing Company, 1989.

[60] Ronald A. Fisher, *Statistical Methods for Research Workers*, Hafner Publishing Company, New York, 1973.

[61] Herve Bourlard, Hynek Hermansky, Nelson Morgan, "Towards increasing speech recognition error rates", *Speech Communication*, 18:205-251, 1996.

[62] Christophe Ris, Stphane Dupont, "Assessing local noise level estimation methods: Application to noise robust ASR", *Speech Communication*, 34(1-2):141-158, April 2001.

[63] Jean-Claude Junqua, *Robust Speech Recognition in Embedded Systems and PC Applications*, Kluwer Academic Publishers, Boston, 2000.

[64] David Arthur Gethin Williams, *Knowing what you don't know: Roles for Confidence Measures in Automatic Speech Recognition*, University of Sheffield, Phd Thesis,May 1999.

[65] Erhan Mengusoglu, Chridtophe Ris, "Use of Acoustic Prior Information for Confidence Measure in ASR Applications", *Eurospeech*, 2001.

[66] C. Uhrik, W. Ward, "Confidence Metrics Based on n-gram Language Model Backoff Behaviors", *Eurospeech*, 1997.

[67] Forsythe, Malcolm and Moler, "Computer Methods for Mathematical Computations", Prentice-Hall, 1976.

[68] Kadri Hacioglu, Wayne Ward, "A Concept Graph Based Confidence Measure", *ICASSP*, 2002.

[69] A. Wendemuth, G. Rose, J. Dolfing, "Advances in confidence measures for large vocabulary", *ICASSP*, 1999.

[70] Olivier Pietquin, Steve Renals, "ASR System Modeling For Automatic Evaluation And Optimization of Dialogue Systems", *ICASSP*, 2002.

[71] Gabriel Skantze, "The use of speech recognition confidence scores in dialogue systems", *GSLT: Speech Technology 5p (HT02) course*, Term paper, Graduate School of Language Technology, Faculty of Arts, Göteborg University, 2003.

[72] R. C. Rose, H. Yao, G. Riccardi, J. Wright, "Integration of utterance verification with statistical language modeling and spoken language understanding", *Speech Communication*, 34:312-331, 2001.

[73] Trausti Thor Kristjansson, "Speech Recognition in Adverse Environments: A Probabilistic Approach", University of Waterloo, Phd Thesis, 2002.

[74] Christophe Couvreur, "Environmental Sound Recognition: A Statistical Approach", Faculté Polytechnique de Mons, Phd Thesis, 1997.

[75] Stéphane Dupont, "Edute et développement d'architectures multi-bandes et multimodales pour la reconnaissance robuste de la parole", Faculté Polytechnique de Mons, Phd Thesis, 2000.

[76] B. L. Pellom, J.H.L. Hansen, "A Duration-Based Confidence Measure for Automatic Segmentation of Noise Corrupted Speech", *International Conference on Spoken Language Processing (ICSLP)*, 1998.

[77] J. Pitrelli, C. Fong, S. Wong, J. Spitz, H. Leung, "Phonebook: A Phonetically Rich Isolated Word Telephone Speech Database", *ICASSP*, 1995.

[78] Jean-Franois Bonastre, Frédéric Bimbot, Louis-Jean Boë, Joseph P. Campbell, Douglas A. Reynolds, Ivan Magrin-Chagnolleau, "Person Authentication by Voice: A Need for Caution", *Eurospeech*, 2003.

[79] C. H. Lee, C. H. Lin, B. H. Juang, "A study on speaker adaptation of the parameters of continous density hidden markov models", *IEEE Transactions on Signal Processing*, 39:806-814.

[80] C. J. Leggetter, P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models", *Computer Speech and Language*, 9:171-185.

[81] Erhan Mengusoglu, "Confidence Measure Based Model Adaptation for Speaker Verification", *The 2nd IASTED International Conference on Communications, Internet & Information Technology*, November 17-19, Scottsdale, AZ, USA, 2003.

[82] R. O. Duda, P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New Jersey, 1973.

[83] P. Nguyen, "Fast Speaker Adaptation", *Technical Report*, Eurecom, 1998.

[84] D. Petrovska, J. Hennebert, H. Melin, D. Genoud, "Polycost: A Telephone-Speech Database for Speaker Recognition", *Speech Communication*, 31(2-3):365-270, 2000.

[85] András Zolnay, Ralf Schlüter, Hermann Ney "Robust Speech Recognition Using a Voiced-Unvoiced Feature", *ICSLP 2002*.

[86] The Speech Training and Recognition Unified Tool (STRUT), Signal Processing and Circuit Theory (TCTS), Faculté Polytechnique de Mons (FPMs), *http://www.tcts.fpms.ac.be/asr/project/strut/*, 1996-2004.

[87] Roni Rosenfeld, "Carnegie Mellon University(CMU) Statistical Language Modeling Toolkit", *http://mi.eng.cam.ac.uk/ prc14/toolkit_documentation.html*, 1994-2004.