

# USING MULTIPLE CODEBOOKS FOR TURKISH PHONE RECOGNITION

Erhan Mengüšođlu  
Hacettepe University  
Dept. of Computer Sciences  
and Engineering  
06532- Beytepe/ANKARA  
TURKEY  
mengus@hun.edu.tr

Harun Artuner  
Hacettepe University  
Dept. of Computer Sciences  
and Engineering  
06532- Beytepe/ANKARA  
TURKEY  
artuner@hun.edu.tr

## Abstract

There are several advantages of speech recognition. It is easier to use speech for data entrance than other tools. It allows to write user-friendly data entrance programs. There are several difficulties in speech recognition. One of these difficulties is noise. Variability in speech is another problem. Even the speech of same speaker varies. because converting speech to text will improve the human-computer interaction, this task is very challenging.

In this paper we proposed a method using multiple codebooks for Turkish Phoneme Recognition. We used a phoneme based speech recognition task, because Turkish Language is phoneme based. We have observed that using one codebook is not sufficient for accurate phoneme recognition. In this paper we have shown that using two codebooks would improve the recognition accuracy in a phoneme recognition task for Turkish Language.

## Keywords

Multiple Codebooks, Phoneme Recognition, Speech Recognition, Turkish Speech Recognition, Self-Organizing Maps, Learning Vector Quantisation.

## 1. Introduction

Speech recognition deals with conversion of speech data to a limited number of symbols. In other terms, the purpose of speech recognition is to obtain text from speech. There are several advantages of speech recognition. It is easier to use speech for data entrance than other tools. It allows writing user-friendly data entrance programs. There are several difficulties with speech recognition. One of these difficulties is noise. Variability in speech is another problem. Even the speech of same speaker varies. Because converting speech to text will improve the human-computer interaction, this task is very challenging.

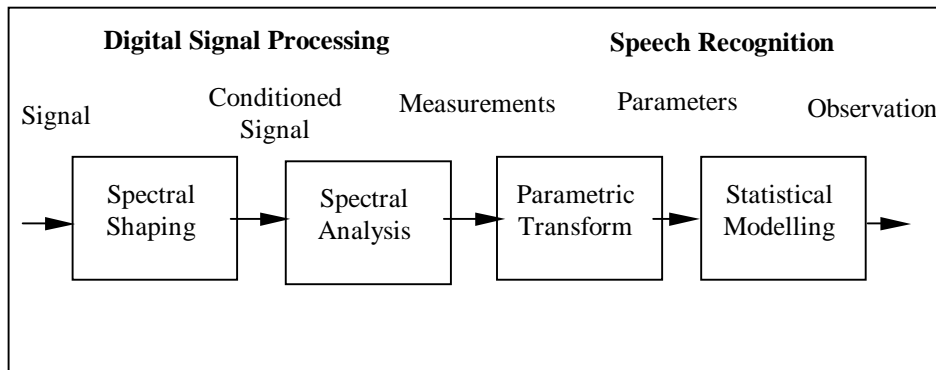
According to the speech unit to be recognized we can classify speech recognition systems as word recognition systems and phone recognition systems. Here we use phone recognition. Phone is the smallest unit of speech. Phone can be represented with

a symbol. This symbol is phoneme. There can be several phones indicating the same phoneme.

After phonemes have been extracted it is easy to convert them to letters, then to words. The performance of the overall system depends on performance of phone recognition. Here we propose a phoneme recognition system for Turkish language. Since Turkish is a phoneme-based language like Finnish or Japanese, phonemes are the same as letters. The second chapter of this paper is about the speech recognition model and methods used in the paper. The third chapter is about the experimental processes. Fourth chapter proposes the results.

## 2. Methods

The first step in speech recognition is parameter extraction from the analog speech signal. The overall speech recognition process consists of spectral shaping, spectral analysis, parametric transformation and statistical modeling phases. This model, which was proposed in Picone, 1993, is as shown in Figure 2.1.



### Spectral Shaping

The very first step of speech recognition is spectral shaping. Spectral shaping includes A/D conversion and digital filtering. A/D conversion is conversion of analog speech signal to digital representation of it. This step is very important because speech recognition tasks need noise-free speech (speech with high SNR-Speech to Noise Ratio-value). It is better to use noise-free speech than using noise cancellation methods to exclude noise from speech. But obtaining noise-free speech is more useless in practical applications. Here we used noise-free speech for experimental purposes, with a sampling frequency of 16 kHz.

After A/D conversion there are some unwanted frequencies in speech signal. Digital filtering emphasizes the important frequency components of speech signal. The most often used filtering technique is a Finite Impulse Response (FIR) filter known as **preemphasis filter**.

$$H_{pre}(z) = 1 + a_{pre}z^{-1} \quad (1)$$

Because hearing is more sensitive to this region, this filter amplifies the above 1-kHz frequency region of the speech signal.

### Spectral Analysis

The second step in model proposed (Figure 2.1) is Spectral Analysis. Spectral analysis is used for some measurements on conditioned speech signal. These measurements are power and spectral amplitude. Power is the temporal power of the speech signal. It can be calculated on a window of speech signal.

$$P(n) = \frac{1}{N_s} \sum_{m=0}^{N_s-1} \left( w(m) s \left( n - \frac{N_s}{2} + m \right) \right)^2 \quad (2)$$

Here  $n$  is the index number of current frame,  $s(n)$  is the signal,  $w(m)$  is the windowing function and  $N_s$  is the number of samples in the window. Window function is used for enhancing the temporal speech signal. There are many types of window functions. The most used window functions are Rectangular Window and Hamming Window. In Rectangular window, each sample in the current frame of speech signal is multiplied with a constant value. In Hamming window each speech sample in the current frame is multiplied by a value calculated from following equation.

$$h(n) = 0.54 - 0.46 \cos\left(\frac{2n\pi}{N-1}\right) \quad (3)$$

Here  $n$  is the index number of samples in the current frame and  $N$  is the total number of samples in current frame. The purpose of this window is to weigh the samples in the middle of the frame.

Power can be computed on frames. Frame duration can be determined experimentally. Extracting useful parameters from speech signal is related to choosing proper frame duration. Because speech signal is continuous, if we take one frame after another it is possible to cause loss of information. So we must use overlapped frames. In this paper we used 16 ms frame duration and 8 ms frame overlaps. This means power values will be calculated for 16 ms of speech signal each time and the next frame will be started from 8<sup>th</sup> ms of previous frame.

Power values cannot give enough information about speech signal. We need to know the frequency content of the signal to parameterize it. Computing spectral amplitude can do this. Spectral amplitude is a measure of power over particular frequency intervals in the spectrum of signal. Taking Fast Fourier Transformation (FFT, Brigham 1974) of speech signal or using Linear Prediction (LP, Atal and Hanauer, 1971) methods can do this simply. There are several other methods for spectral amplitude computation. Methods can be divided into two groups: DFT (Discrete Fourier Transformation) based and LP based. Both of them try to find best parameters for recognition of speech signal.

### Parametric Transform

Speech parameters can be generated by some measurements on speech signal. Here, the aim is to find a parametric estimation for speech signal. There are several methods for parametric transform. Parametric transformations are based on spectral shaping. For example mel-cepstrum method (Davis and Mermelstein, 1980), which uses DFT and IDFT (Inverse DFT) for feature extraction.

Here we used a Linear Prediction derived spectral amplitude computation method called RASTA (RelAtive SpecTrA) proposed by Hermansky and Morgan 1994. This method is based on modeling of environmental effects, noise, in speech signal. Its previous version is Perceptual Linear Prediction (PLP) by Hermansky 1990. PLP first uses Fourier transformation, applies a Linear Prediction model. RASTA changes filters used in PLP by filters with sharp spectral zero at zero frequency. The purpose is to suppress any slowly varying frequency components from speech signal.

### Statistical Modeling

Statistical modeling of speech consists of some statistical processes for robust speech-to-text conversion. It is based on some signal observations. The most used statistical modeling method is Hidden Markov Model (Rabiner, 1989). HMM is based on some stochastic processes on speech. There is a probabilistic finite-state model for each speech unit.

In this paper we used a Neural Network based model for classification of speech units. This method is Learning Vector Quantization (LVQ). It is based on Self Organizing Maps of Kohonen (Kohonen, 1990). LVQ is based on Nearest Neighbor algorithm. It results a mapping of observed vectors called codebook. Then this codebook is used for classification.

### **3. Experiments**

In this paper we used the speech recognition system proposed in Mengüšoğlu, 1999. The schematic representation of the speech recognition system used for experiments can be seen on Figure 3.1.

The first step of experimental procedure is corpus preparation. Because we used a phoneme-based approach for Speech Recognition, we need to represent all phonemes of Turkish Language in corpus. Each phoneme has different allophones according to its location in the word and in the syllable (Artuner, 1994). So we need to sample each of these allophones in the corpus. Corpus contains 248 words. There are 29 letters in Turkish Language. Because the language is phoneme based, we can take each letter as a phoneme. There are 8 voiced and 21 unvoiced phonemes. According to the position of the phoneme in the word and syllable we proposed that there are 5 allophones for voiced and 4 allophones for unvoiced phonemes. The details of this procedure can be seen on Table 3.1.

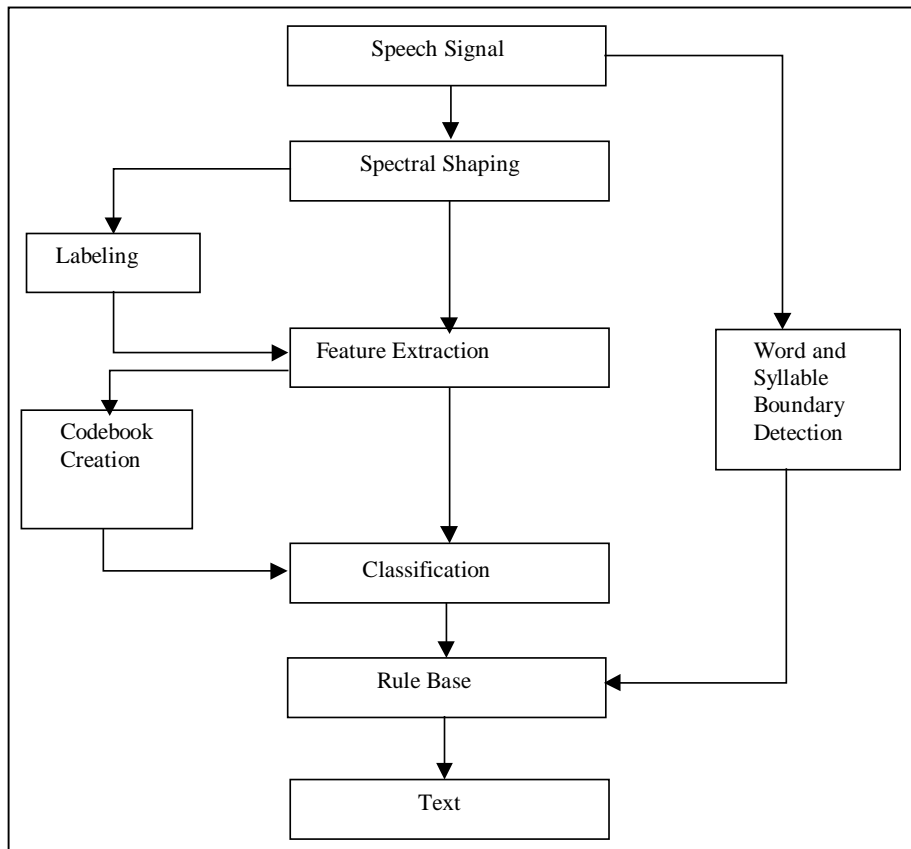


Figure 3.1. Schematic Representation of Speech Recognition System.

The next step is speech data preparation. The speech recognition system used here is speaker dependant. One speaker has uttered each word in the corpus only once. There are 248 words in the corpus. Speech has been recorded in a silent room using 48kHz frequency. Recording media was ADAT (an eight-track digital tape recorder by Alesis - Alesis Digital Audio Tape). After recording the speech was digitized in 44kHz frequency and 16 bit amplitude values using a sound card and a mixer. We used 16kHz speech in experiments, so the speech data then converted to 16kHz and 16bit format.

Table 3.1. Word selection for corpus.

Location	First Syllable			After First Syllable		
	First	Middle	Last	First	Middle	Last
Voiced Phonemes	2	2	2	-	2	2
Unvoiced Phonemes	2	-	2	2	-	2

Now we have digitized speech data. There are three concurrent processes here. Labeling of speech, Feature extraction and Boundary Detection.

Labeling process is needed for codebook creation. This process is for training purposes. Because the spectral content of speech helps us to determine the phoneme boundaries, we need to visualize the spectral contents of speech while labeling it. Each phoneme was labeled with its corresponding letter. Plosives (P, Ç, T, K) have been labeled with two labels, one for unvoiced part and one for voiced part.

Feature extraction is the most important step of speech recognition. The performance of speech recognition system is closely related to this step. We used RASTA (Hermansky and Morgan, 1994) method for feature extraction here. RASTA method was also prepared as a program package by authors. The command line for feature extraction by RASTA is as follows:

```
rasta -a -A -i speech_data.asc -o speech_data.rasta -S 16000 -m 21 -w 16 -s
```

8

Here the input data is *speech\_data.asc* and the features are returned in *speech\_data.rasta*.

Boundary detection process is for detection of word boundaries. Although we used isolated words we need to determine the start and end points of the words. This process can be described as discrimination of silence from speech. We used a method using zero-crossing rates and energy values for word boundary detection. (Rabiner and Schafer, 1978). We used Hamming windowed speech data for zero crossing and energy calculations.

The next step is codebook creation. Codebook can be defined as a file containing the reference templates of all phonemes. Here we used Learning Vector Quantization algorithm of Kohonen et. al. (Kohonen et. al., 1996). The vectors used here are the feature vectors obtained by RASTA methods. First feature vectors were labeled and then LVQ is performed. The result of this process is codebook.

The main purpose was using two codebooks was speech recognition. One of them was used for phoneme discrimination and the other for voiced-unvoiced discrimination. In this paper we used both of them concurrently.

Here we did not use parallel processing for labeling process, but it will be better to use parallel processing techniques for more efficient systems.

The codebook used for phoneme discrimination was obtained by following way:

- Feature vectors extracted by RASTA labeled with phoneme labels
- LVQ process was applied

The codebook used for voiced-unvoiced detection was obtained by following way:

- Each feature vector labeled as voiced or unvoiced according to its phoneme label
- LVQ process was applied.

Codebook creation followed by classification. Original feature vectors extracted from speech was classified by using both of codebooks. The results then compared for advantage of using two codebooks.

#### 4. Results

Experiments showed that using two codebooks concurrently for phoneme recognition has improved the recognition accuracy.

The result of classification is a label array. This label array then used for determining the text equivalent of speech. There are two label arrays here. First one is the result of

classification using phoneme codebook and the second one is the result of classification using voiced/unvoiced codebook. After obtaining label arrays, we applied an enhancement algorithm on these arrays. This algorithm is as follows:

- Start from first label of array,
- Find the most frequent label from within next five label,
- Change the current label by this label,
- Continue this operation until reaching the end.

This algorithm has been applied to both of the label arrays. Then to find the text equivalent of speech those two label arrays used by the following algorithm:

- Start from first label of phoneme-labeled array,
- Cross-check the voiced/unvoiced-labeled array,
- If there is an unvoiced label which is labeled as voiced in the second array (or vice versa) then discard it,
- Continue this operation until reaching the end.

The following results have been observed from experimental processes.

Results for using one codebook (codebook by phoneme-labeled vectors):

Original	a, an, kaplan, baba, savaS, koca, aC, lamba, mala, Irmak, flrCa
Resulted	ak, an, kaplan, baba, savaSj, koca, tkaaC, lamba, mala, Irmak, flrkCa
Original	be, ben, kablo, hibe, teSebbUs, bilgi, kobra, kalbe, lakab
Resulted	be, ben, kablo, hiUe, teSebUs, bilgi, kobra, kalbe, lakab
Original	ce, cin, sac, acI, topac, ceviz, secde, kucak, teveccUh
Resulted	dce, bcin, sact, acI, topackCt, ievizs, sece, kuak, tecUh
Original	de, demir, ad, kadI, Ustad, dUz, gaddar, zorda, teyid
Resulted	de, deir, adt, kadI, Ustadd, dUz, gbgadar, zorda, teyid

Results for using two codebooks concurrently:

Original	a, an, kaplan, baba, savaS, koca, aC, lamba, mala, Irmak, flrCa
Resulted	a, an, kaplan, baba, savaS, koca, aC, lamba, mala, Irmak, IrkCa
Original	be, ben, kablo, hibe, teSebbUs, bilgi, kobra, kalbe, lakab
Resulted	be, ben, kablo, hiUe, teSebUs, bilgi, kobra, kalbe, lakab
Original	ce, cin, sac, acI, topac, ceviz, secde, kucak, teveccUh
Resulted	ce, bcin, sackc, acI, topac, ieeiz, sece, kuak, tecUh
Original	de, demir, ad, kadI, Ustad, dUz, gaddar, zorda, teyid
Resulted	de, demir, ad, kadI, Ustad, dUz, gadar, zorda, teyid

As can be seen from the results above, there are some improvements in results of using two codebooks.

## 5. Conclusion

Using multiple codebook technique presented in this paper is a LVQ based technique. We showed that using multiple codebooks would improve the recognition accuracy in a LVQ based speech recognition system. Another advantage of using multiple codebooks

is that it makes concurrent processing possible. Since the speed of labeling process is important for efficient speech recognition, it will be better to use parallel processing for phoneme labeling. So, more efficient speech recognition systems can be implemented. The number of codebooks can be increased. We suggest that if the number of codebooks increased there will be more accurate speech recognition tasks. For example one can use another codebook for phone-center and transition classification.

## References

1. Picone, W. J., 1993. *Signal Modeling Techniques in Speech Recognition*
2. Brigham, O. E., 1974. *The Fast Fourier Transform*. Englewood Cliffs, NJ: Prentice Hall.
3. Atal, B. S., Hanauer, S. L., 1971. Speech Analysis and synthesis by linear prediction of the speech wave. *J. Acoust. Soc. America*. Vol. 50, No. 2, pp. 637-655.
4. Davis, S. B., Mermelstein, P., 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 28, no. 4, pp. 357-366.
5. Hermansky, H., 1990. Perceptual Linear Predictive (PLP) Analysis of Speech. *Journal Of Acoustic Society of America*. vol. 87, no. 4. pp. 1738-1752.
6. Hermansky, H., Morgan, N., 1994. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578-589.
7. Rabiner, L. R., 1989. A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of IEEE*, vol. 77, no. 2, pp. 257-286.
8. Kohonen, T., 1990. The self-organizing map. *Proceedings of IEEE*, vol 78, pp. 1464-1480.
9. Artuner, H., 1994, *Bir Türkçe Fonem Kümeleme Sistemi Tasarımı ve Gerçekleştirimi.*, PhD thesis.
10. Rabiner, L., R. Schafer, R., W., 1978. *Digital processing of speech signals*. Prentice Hall Inc.
11. Kohonen, T., et. al., 1996, *LVQ\_PAK: The Learning Vector Quantization Program Package*, Report A30.
12. Kohonen, T., et. al., 1996, *SOM\_PAK: The Self Organizing Map Program Package*, Report A31.
13. Mengüsoğlu, E., 1999, *Rule Based Design and Implementation of a Speech Recognition System for Turkish Language*. Master Thesis.