

Dinamik Zaman Bükmesi Yöntemiyle Hece Tabanlı Konuşma Tanıma Sistemi

Rıfat AŞLIYAN¹, Korhan GÜNEL², Tatyana YAKHNO¹

¹ Dokuz Eylül Üniversitesi, Bilgisayar Mühendisliği Bölümü, İZMİR

² Adnan Menderes Üniversitesi, Matematik Bölümü, AYDIN
rasliyan@adu.edu.tr, kgunel@adu.edu.tr, yakhno@cs.deu.edu.tr

Özet: Bu çalışmada, hece tabanlı Türkçe konuşma tanıma sistemi geliştirilmiştir. Türkçe, dünya dilleri arasında yapı yönüyle sondan eklemeli diller grubuna girdiğinden bir sözcükten birçok farklı sözcük türetilbilmesine neden olmaktadır. Bundan dolayı sözcük tabanlı sistem yerine sınırlı sayıda hecelere sahip Türkçe için hece tabanlı sistem tercih edilmiştir. Bu çalışmamızda, konuşma tanıma sistemlerinde en çok kullanılan yöntemlerden biri olan dinamik zaman bükmesi kullanılmıştır. Sistemin uygulanmasında kullanmak amacıyla imla kılavuzundan birbirine benzer 200 sözcük seçilmiştir. Bu sözcüklerin her biri için 25'er tane wave formatında ses dosyaları kaydedilmiştir. İlk olarak, sözcük ses sinyallerinden hece sınırlarının tespiti algoritmasıyla hecelerin başlangıç ve bitiş sınırları tespit edilmiştir. Daha sonra her hecenin için doğrusal öngörülü kodlama(LPC), parcor, cepstrum ve MFCC öznitelikleri çıkarılmıştır. Bu öznitelikler kullanılarak her hece için bir medyan hece ses sinyali seçilmiştir. Dinamik zaman bükmesi yöntemi kullanılarak benzer sözcükteki hecelere en çok benzeyen medyan heceler sıralı bir şekilde tespit edilir. En çok benzeyen hecelerden 5 tanesi seçilerek tekrar, bu hecelerin bütün ses şablonları arasından tanıma işlemi gerçekleştirilir. Her bir hece için en çok benzeyen 5 hece sıralı bir şekilde tespit edilir. Heceler art arda eklenerek sözcük oluşturulur. Eğer bu sözcük sözlüğümüzde yok ise bir sonraki heceler eklenerek yeni bir sözcük türetilir. Eğer sözlükte var ise tanınan sözcük bu olacaktır. Test işlemi için 2000 tane sözcük kaydedildi ve sistemin mfcc özniteliklerine göre doğru tanıma oranı %95,3 olmuştur.

Anahtar Sözcükler: Dinamik Zaman Bükmesi, DTW, Konuşma Tanıma, Hece Tabanlı Konuşma Tanıma.

Syllable Based Speech Recognition Using Dynamic Time Warping

Abstract: In this study, we have developed a syllable based speech recognition system using Dynamic Time Warping(DTW). As it is known, Turkish is agglutinative language, so many different Turkish words can be generated from a Turkish word by adding suffixes. For Turkish speech recognition, syllables as speech units are more adequate than words. 200 similar words are chosen from Turkish spelling dictionary to implement the system. For each word in the dictionary 25 word sound signals are recorded in wave file format. Firstly, syllable onsets are decided on the word sound signals by syllable onset detection algorithm. After that, feature vectors of linear predictive coding (lpc), parcor, cepstrum and mel frequency cepstral coefficient (mfcc) are derived from the signals. According to these features, a median feature is found for each syllable. The most similar median syllables are determined by DTW and sorted by ascending order. The most similar 5 syllables are chosen and the recognition operation is done again, but for all these 5 syllable sound signal templates. The best matched syllables in the word is

added in order and a word is produced. If this word is in the dictionary, the recognized word of the system is this one. Otherwise, producing the words from the syllables are continued in the most similar way. There are 2000 words for testing the system and the accuracy result is 95.1% according to mfccr features.

Keywords: Dinamic Time Warping, DTW, Speech Recognition, Syllable Based Speech Recognition.

1. Giriş

Konuşma tanıma, sözcük ses sinyallerinin işlenerek metne dönüştürülmesi işlemidir. Konuşma tanıma üzerine yapılan çalışmalar son 25 yılda hızlanarak artmıştır. Bu çalışmaların büyük çoğunluğu konuşma tanımadaki temel bileşen olarak fonem ve sözcük birimleri kullanılmıştır. Fonem benzeri birimler arasındaki sınırların tespit edilmesi oldukça zor bir süreçtir. Sözcük birimlerini temel alan sistemler, fonem birimlerini kullanan sistemlerin problemlerini içermese de bir çok hesaplama ve veri eğitimini beraberinde getirmektedir. Türkçe sondan eklemeli diller [5] grubuna girdiğinden geniş ölçekli konuşma tanıma sistemlerinde sözcüğü temel birim olarak almak problem oluşturabilir. Çünkü, bir sözcüğe ekler getirmek suretiyle onlarca yeni sözcük türetilir. Bu yüzden çalışmamızda sistemin temel birimi olarak fonem ve sözcük arasında bir birim olan hece seçilmiştir. Günümüze kadar yayınlanmış heceleri temel alan konuşma tanıma [3], [11], [12] sistemleri çok fazla değildir. Konuşma tanıma sistemleri konuşmacıya bağımlı veya konuşmacıya bağımsız olabilir. Konuşma tanımadaki yaygın olarak tercih edilen öznitelikler doğrusal öngörülü kodlama (lpc) [1], [2], [9], [10], [12], parcor ve cepstrum [8], [12] ve mfcc (mel frequency cepstral coefficient) öznitelikleridir. En çok kullanılan yöntemler dinamik zaman bükmesi [4], [6], [7], yapay sinir ağları ve saklı markov modelidir [5], [12], [13].

Bu çalışmada dinamik zaman bükmesi yöntemi kullanılarak ayrıık sözcük tanıma uygulama

ması gerçekleştirilmiştir. İmla kılavuzundan birbirine çok benzer 200 sözcük seçilerek lpc, parcor, cepstrum ve mfcc öznitelik değerleri kullanılarak kişiye bağımlı uygulamalar yapılmıştır. Test işlemi için 2000 tane sözcük kaydedildi ve sistemin en başarılı olduğu mfcc özniteliklerine göre doğru tanıma oranı %95,1 olmuştur.

Bu çalışmanın ikinci bölümünde sistem mimarisi hakkında genel bilgiler verilmiştir. Üçüncü bölümünde ise hece sınırlarının nasıl tespit edildiği hakkında bahsedilmiştir. Dördüncü bölümde kullanılan özniteliklerin hece ses sinyallerinden nasıl çıkarıldığı anlatılmaktadır. Beşinci bölümde artışleme algoritması verilmiştir. Altıncı ve yedinci bölümde sırasıyla sistemin test edilmesi, tartışma ve sonuç hakkında bahsedilmektedir.

2. Sistem Mimarisi

Bu çalışma Şekil 1’de görüldüğü üzere genel anlamda dört aşamadan oluşmaktadır. Birinci aşamada 200 sözcüğe ait hecelerin öznitelik şablonları veritabanı oluşturulur. İkinci aşamada her bir hece şablonundaki medyan hece öznitelik vektörleri tespit edilir. Üçüncü aşamada ise DTW [4] yöntemiyle hece tanıma işlemi gerçekleştirilir. Son aşamada tanımayı iyileştirmek için artışleme (postprocessing) uygulanır.

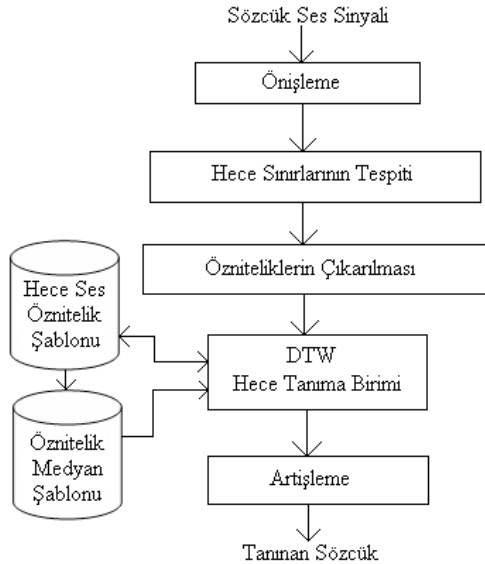
Her bir hecenin öznitelik şablonlarının hesaplanması için sözcükler wave formatında 2 saniyelik zaman aralığında bir konuşmacı tarafından 11025 Hz’de örneklenip 16 bit darbe

kod kiplenimi (PCM) kullanılıp nicemlenerek ve önişleme işleminden geçirilmiştir. Önişlemede ses sinyallerinin ortalaması sıfır olacak şekilde yeniden düzenlenmiştir (1). x_n ses sinyalini, m ses sinyalinin ortalamasını ve y_n yeni ses sinyalini temsil etmektedir.

$$y_n = x_n - m \quad (1)$$

$$m = \left(\sum_{i=1}^k x_i \right) / k$$

Hece sınırları tespit edilmeden ve öznelilik çıkartma işlemi yapılmadan önce ses örneklerine önvurgulama yapılmaktadır. Daha sonra hece sınırları tespit edilir (Bölüm 3). Her hecenin ses örnekleri 20 ms'lik çerçevelere ayrılarak çerçeveler üzerine Hamming pencereleme uygulanır. Çerçeveler arasındaki örtüşme 10 milisaniyedir. Daha sonra Bölüm 4'de anlatıldığı gibi hecenin her çerçevesi için öznelilik değerlerinden 8'er tane lpc, parcor, cepstrum ve mfcc hesaplanır.



Şekil 1. Sistemin Genel Yapısı

Sistem dinamik zaman bükmesi yöntemini kullanarak heceler en benzer olanlarını hece öznelilik medyan şablonlarına göre elde eder ve sıralar. Tanınacak olan sözcüğün hecelerinin en benzer 5 tanesi tekrar tanıma işleminden geçirilir. Fakat bu sefer beş hece arasından daha ayrıntılı tanıma sürecinden gerçekleştirilir. Yani Hece Ses Öznelilik Şablonu veritabanından tanıma işlemi yapılır. Dolayısıyla bu beş heceden en çok hangisine benzer olduğu tespit edilir. Artışleme (Bölüm 5) kullanılarak tanıma işlemi sonlandırılır.

Çalışmalarımız Pentium Centrino 1.6 işlemci, 768 RAM, 40 GB sabit disk, Windows XP işletim sistemi, ses kartı, hoparlör ve mikrofona sahip bilgisayar üzerinde gerçekleştirilmiştir. Uygulama programları Matlab 6.5 ile kodlanmıştır.

3. Hece Sınırlarının Tespiti

Hece sınırları tespit yöntemi iki aşamadan oluşmaktadır. Birincisi, sözcüğün başlangıç ve bitiş noktalarının belirlenmesi işlemidir. İkincisi, sözcükteki heceler sınırlarının tespiti işlemidir. Aşağıda her ikisinin algoritması verilmiştir.

3.1 Sözcük Başlangıç ve Bitiş Noktası Tespiti Algoritması

1. X , ses dosyasını temsil eden vektördür. Çalışmamızda $N = 22050$ dir.

$$x = (x_1, x_2, x_3, \dots, x_N) \quad (1)$$

2. l , x vektörünün ilk 200 değerinin ortalamasıdır. \tilde{x} vektörü, x vektörünün yatay eksenine odaklanmış halidir.

$$l = \left(\sum_{i=1}^{200} x_i \right) / 200 \quad (2)$$

$$\tilde{x} = x - 1 = (x_1 - 1, \dots, x_N - 1) = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N) \quad (3)$$

3. M : \tilde{x} vektörünün maksimum değeridir.

I : \tilde{x} vektörünün maksimum değerinin indeksidir.

E_b ve E_s : Srasıyla başlangıç ve son eşik değerleridir.

$$[M, I] = \max(\tilde{x}) \quad (4)$$

$$E_b = M / I, \quad E_s = M \cdot (N - I) \quad (5)$$

4. \tilde{x} vektörü, 350 örnekten oluşan pencerelere bölünür. \bar{x} vektörü, 350 örnekli her pencerele-
rin ortalamasının vektörüdür.

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_p) \text{ ve } p = N / 350 \quad (6)$$

$$\bar{x}_i = \left(\sum_{k=i*350}^{(i+1)*350-1} \tilde{x}_k \right) / 350, \quad i=1, 2, \dots, p \quad (7)$$

5. $i=1, 2, \dots, p-1$ için,

$$\bar{x}_{E_i} = (\bar{x}_{E_1}, \bar{x}_{E_2}, \dots, \bar{x}_{E_{p-1}}) \text{ ve } \bar{x}_{E_i} = \bar{x}_{i+1} / \bar{x}_i \quad (8)$$

6. S_b , sesin başlangıç indeksidir.

For $r=1$ To $p-1$

Eğer $\bar{x}_{E_r} > E_b$ ise $S_b = r * 350$

End

7. S_s , sesin bittiği indekstir.

For $r = x_{E_{p-1}}$ DownTo 1

Eğer $1 / \bar{x}_{E_r} > E_s$ ise $S_s = r * 350$

End

8. Algoritmanın 6. ve 7. adımlarında sesin başladığı ve bittiği indeksler yaklaşık olarak tespit edilir. Tam olarak belirlemek için sıfır keşişinin olduğu indeksler tespit edilir.

$\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N)$ ses vektörünü kullanarak sıfır keşişi vektörü $z = (z_1, z_2, \dots, z_{N-1})$ elde edilir.

For $k=2$ To N

Eğer $\tilde{x}_{k-1} / \tilde{x}_k < 0$ ise $z_{k-1} = 1$

Değilse $z_{k-1} = 0$

End

9. Art arda sıfır keşişi noktaları arasında olan uzaklıklar hesaplanarak yeni bir sıfır keşişi uzaklıkları vektörü $\tilde{z}_k = (\tilde{z}_1, \tilde{z}_2, \dots, \tilde{z}_{N-1})$ bulunur.

For $k=1$ To $N-1$

Eğer $z_k = 1$ ve k 'dan sonra gelen indekslerden değeri 1 olan ilk indeks h ise

$$\tilde{z}_k = h - k$$

Aksi halde $\tilde{z}_k = 0$

Eğer $z_k = 0$ ise $\tilde{z}_k = 0$

End

10. Sıfır keşişi eşik değeri, $T=100$ olarak kabul edildi.

SB : En son olarak sesin başladığı indeks değeri olacaktır.

$$\mathfrak{B} = S_b$$

For $k = S_b$ DownTo 1

Eğer $\tilde{z}_k > 0$ ve $\tilde{z}_k < T$ ise $\mathfrak{B} = k$

Eğer $\tilde{z}_k = 0$ ise döngüye devam et.

Eğer $\tilde{z}_k > T$ ise döngüden çık.

End

11. SS : En son olarak sesin bittiği indeks değeri olacaktır.

$$\mathfrak{S} = S_s$$

For $k = S_s$ To N-1

Eğer $\tilde{z}_k > 0$ ve $\tilde{z}_k < T$ ise $\mathfrak{S} = k$

Eğer $\tilde{z}_k = 0$ ise döngüye devam et.

Eğer $\tilde{z}_k > T$ ise döngüden çık.

End

3.2 Sözcükteki Hecelerin Sınırlarının Tespiti Algoritması

Sesin başlangıç ve bitiş indeksleri (SB ve SS) tespit edildikten sonra aşağıdaki algoritmayla hece sınırları tespit edilir.

$$1. n = (n_1, n_2, \dots, n_k) = (\tilde{x}_{\mathfrak{B}}, \tilde{x}_{\mathfrak{B}+1}, \dots, \tilde{x}_{\mathfrak{S}})$$

2. n vektörü örtüşme yapılmaksızın 900 örneklili pencereye bölünür. \bar{n} vektörü, 900 örneklili her pencerenin ortalamasının vektörüdür.

$$\bar{n} = (\bar{n}_1, \bar{n}_2, \dots, \bar{n}_p) \quad \text{ve} \quad p = k/900 \quad (9)$$

$$\bar{n}_i = \left(\sum_{m=i*900}^{(i+1)*900-1} n_m \right) / 900, \quad i=1,2,\dots,p \quad (10)$$

3. \bar{n} vektörünün art arda gelen değerleri arasında eğimler hesaplanıp eğim vektörü oluşturulur. $i=1,2,\dots,p-1$ için,

$$\bar{n}_E = (\bar{n}_{E_1}, \bar{n}_{E_2}, \dots, \bar{n}_{E_{p-1}}) \quad \text{ve} \quad \bar{n}_{E_i} = \bar{n}_{i+1} / \bar{n}_i \quad (11)$$

4. Eğim vektöründen +1 ve -1'lerden oluşan

yeni bir vektör $a = (a_1, a_2, \dots, a_{p-1})$ hesaplanır. Yani artan ve azalan vektörü bulunur.

For $k=1$ To $p-1$

Eğer $n_{E_k} \geq 0$ ise $a_k = 1$

Aksi halde $a_k = -1$

End

5. H: Sözcükteki hece sayısı

$$H = 0$$

For $k=2$ To $p-1$

Eğer $a_{k-1} = 1$ ve $a_k = -1$ ise $H = H + 1$

End

6. a vektöründeki -1 değerlerini içeren indeks gruplarının ortasındaki indeks değerleri, yaklaşık olarak hece sınırlarının olduğu değerler olmaktadır. Hece sınırları H-1 tane olacaktır.

Hece sınırları vektörü $s = (s_1, s_2, \dots, s_{H-1})$ hesaplanır. S_i değerleri \tilde{x} vektörünün indekslerini tutan değerlerdir.

For k=1 To H-1

a vektöründe artardagelen k 'inci-1 değerlerine sahip indekslerin ortasındaki indeks w ise

$$s_k = \mathcal{B} + 900 * w$$

End

7. Şimdiye kadar \tilde{x} vektöründe sesin başlangıç değeri SB ve bitiş değeri SS kesin olarak tespit edildi. S vektörü ise heceler arasındaki yaklaşık sınır indeksleri vektörüdür. Daha doğru sınırları bulmak için aşağıdaki işlem yapılır ve

$\tilde{s} = (\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{H+1})$ vektörü elde edilir. Burada

$\tilde{s}_1 = \mathcal{B}$ ve $\tilde{s}_{H+1} = \mathcal{S}$ olmak üzere,

For i=1 To H-1

$s_i - 500$ ile $s_i + 500$ aralığında 20 örnekli pencereler oluşturulur ve bu pencerelerin ortalamaları hesaplandıktan sonra en küçük ortalamaya sahip pencerenin ortasındaki indeks q

ise $\tilde{s}_{i+1} = q$ olur.

End

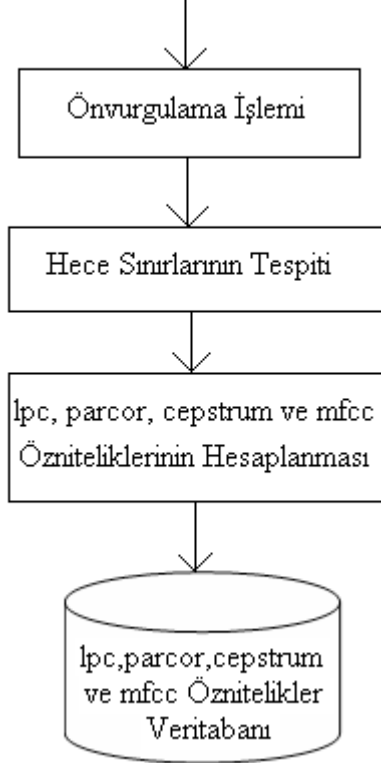
8. \tilde{x} ses vektöründeki hecelerın sınır indeksleri \tilde{s} vektörü şeklinde bulunur. k 'inci hecenin hece başlangıcı \tilde{s}_k ve hece bitişı \tilde{s}_{k+1} indeksi olacaktır. Sözcükte H tane de hece vardır.

4. Hecelerin LPC, PARCOR, CEPSTRUM ve MFCC Özniteliklerinin Hesaplanması

İlk olarak ses vektörü önvurgulama yöntemiyle filtrelenir. Sonra, 20 ms'lik çerçevelere bölünür. 10 ms'lik örtüşme kullanılır. Her çerçeveye Hamming pencereleme uygulanır.

Karşılıklı ilinti [9], [10] işlemiyle özilinti vektörü [9], [10] hesaplanır. Levinson metoduyla [10] doğrusal öngörülü kodlama ve Parcor öznitelikleri çıkarılır. Cepstrum öznitelikleri de hesaplanır. Sonuç olarak, her çerçeve için 8 lpc, parcor, cepstrum ve mfcc öznitelik değerleri elde edilir. Her hece için oluşturulan bu öznitelik vektörleri daha sonra kullanılmak üzere dosya adı, hece ismi ve dosya uzantısı da "fetN" olacak şekilde kaydedilirler. N, hecenin sözcükteki kaçınıcı hece olduğunu ifade eder.

200 Sözcük Wav Ses Dosyası



Şekil 2. Öznitelik çıkarılması işlemi

5. ART İşlem Algoritması

N: Test veritabanından alınan sözcüğün hece sayısı.

$H_k(s)$: Test edilecek sözcüğün k 'inci hecesine s 'inci sırada en çok benzeyen hecedir.

1. $i=1,2,\dots,0$ ve s_i : i 'inci heceye en çok benzeyen 10 heceden biri.

$H_1(s_1)H_2(s_2)\dots H_N(s_0)$ şeklinde heceler birleştirilir ve yeni sözcük oluşturulur. Toplam 0^N sözcük elde edilir.

2. Her sözcük için bir düzey belirlenir. 1. adımdaki sözcüğü oluşturan hecelerın sıralarının toplamı hesaplanır ve bu toplam o sözcüğün düzeyi olur.

3. Sözcükler, sözcük düzeyine göre sıralanır.

4. Sözcük düzeyi en küçük olandan başlanır ve bu sözcük, sözcük veritabanında mevcut ise sözcük bulunmuş olur diğer sözcüklere bakılmaksızın işlem biter. Hiçbir sözcük veritabanında yoksa sistem bir sözcük bulamamıştır.

6. Sistemin Test Edilmesi

Sistemin test edilmesi için 2000 wav ses dosyası içeren test veritabanı oluşturulmuştur. 200 farklı sözcük ses sinyalleri bulunan bu veritabanında her sözcük 10 defa kaydedilmiştir. Buradaki ses dosyalarının hece sınırlarının tespitinden sonra her hecenin LPC, Parcor, Cepstrum ve mfcc öznitelikleri hesaplanmıştır. Test veritabanındaki sözcüklerin her bir hecesine en çok benzeyen 5 tane hece dinamik zaman bükmesi kullanılarak bulunur. Uzaklık değeri en küçük olan heceler en çok birbirine benzeyen hecedir.

Test veritabanındaki sözcüğün hecelerine en küçük uzaklıkta olan heceler birleştirilirse en yakın metin sözcüğü elde edilir. Tablo 1'de kullanılan öznitelikler göre ve art işlem kullanılıp kullanılmadığına bağlı olarak sistemin tanıma oranları verilmiştir. Buna göre artışle-

me kullanılarak tanıma başarısı yaklaşık %10 artmıştır. En büyük başarı mfcc özniteliği üzerinde artışlemlenerek elde edilmiştir ve başarı oranı %95,1 olmuştur.

Yöntem	Öznitelikler			
	lpc	parcor	cepstrum	mfcc
DTW	%64,9	%75,4	%79,9	%90,3
DTW (Artışleme)	%89,1	%89,4	%90,6	%95,1

Tablo 1. Sistemin başarı yüzdeleri.

7. Tartışma ve Sonuçlar

Bu çalışmada, hece tabanlı Türkçe ayırık kişiye bağımlı konuşma tanıma sistemi geliştirilmiştir. Yöntem olarak dinamik zaman bükmesi kullanılmıştır. Öznitelik olarak, doğrusal öngörülü kodlama(LPC), parcor, cepstrum ve mfcc öznitelikleri seçilmiştir ve uygulamalar gerçekleştirilip karşılaştırılmıştır. Sisteme dahil edilen artışleme yöntemi sistemin başarımını oldukça artırmıştır. Sistemin en başarılı özniteliği mfcc olmuştur ve doğru tanıma oranı %95,1 olaral gözlemlenmiştir.

Çalışmanın daha sonraki aşamasında yapay sinir ağları ve saklı markov modeli yöntem olarak kullanılacaktır. Bütün hecelerın kullanılıp modellenmesiyle geniş dağarcıklı Türkçe konuşma tanıma sistemi geliştirilmesi mümkün olacaktır.

8. Teşekkür

Bu çalışma Adnan Menderes Üniversitesi Bilişimsel Araştırma Projeleri (BAP) kapsamında desteklenmektedir.

9. Kaynaklar

- [1]. Harma, A., "A comparison of warped and conventional linear predictive coding", IEEE Trans. Speech and Audio Processing, July 2001.
- [2]. Harma, A., "Linear predictive coding with modified filter structures", IEEE Trans. Speech Audio Processing, 9(8):769 --777, November 2001.
- [3]. Jones, R. J., Downey, S. Mason, J. S., "Continuous Speech Recognition using Syllables", Proc. Eurospeech, Volume 3, pp. 1171-1174, 1997.
- [4]. Kruskall, J. and M. Liberman. The Symmetric Time Warping Problem: From Continuous to Discrete. In Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison, pp. 125-161, Addison-Wesley Publishing Co., Reading, Massachusetts, 1983
- [5]. Mengusoglu, E., and Derro, O., "Turkish LVCSR: Database preparation and Language Modeling for an Agglutinative Language", ICASSP'2001, Student Forum, May 2001, Salt-Lake City.
- [6]. Myers, C. S., Rabiner, L. R. and Rosenberg, A. E., "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition", IEEE Trans. Acous., Speech, and Sig. Processing, Vol. ASSP-28, no. 6, Dec 1980, , pp. 623-635.
- [7]. Paliwal, K. K., Agarwal A. and Sinha, S. S., "A modification over Sakoe and Chiba's dynamic time warping algorithm for isolated word recognition", Signal Processing, Vol. 4, No. 4, pp. 329-333, July 1982.
- [8]. Paliwal, K. K., "On the performance of the quefreny-weighted cepstral coefficients in vowel recognition", Speech Communication, Vol. 1, No. 2, Aug. 1982, pp. 151-154.
- [9]. Proakis, J. G. And Manolakis, D. G., Digital Signal Processing: Principles and Application, Prentice-Hall, Upper Saddle River, NJ, 1996.
- [10]. Rabiner, L. and Juang B. H., Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [11]. Rosenberg, A. E., Rabiner, L. R., Levinson, S. E. and Wilpon, J. G., "A preliminary study on the use of demissyllables in automatic speech recognition", Conf. Rec. Int. Conf. on Acous., Speech, and Sig. Processing, GA, pp. 967-970 Mar 1981, Atlanta.
- [12]. Shafran, I., Clustering wide context and HMM topologies for spontaneous speech recognition, Ph.D. Thesis, University of Washington, 2001.
- [13]. Svendsen, T., Paliwal, K. K., Harborg E. and Husoy, P. O., "A modified acoustic subword unit based speech recognizer", Proc. IEEE Intern. Conf. on Acoustics, Speech and Signal Processing, pp. 108-111, May 1989, Glasgow, U.K.