

Dinamik Zaman Bükmesi ve Çok Katmanlı Algılayıcı Kullanarak Hece Tabanlı Türkçe Konuşma Tanıma

Syllable Based Turkish Speech Recognition using Dynamic Time Warping and Multilayer Perceptron

Rıfat AŞLIYAN¹, Korhan GÜNEL² ve Tatyana YAKHNO¹

1. Bilgisayar Mühendisliği Bölümü, Dokuz Eylül Üniversitesi, İzmir
rasliyan@yahoo.com, yakhno@cs.deu.edu.tr
2. Matematik Bölümü, Adnan Menderes Üniversitesi, Aydın
kgunel@adu.edu.tr

Özetçe

Bu çalışmada, dinamik zaman bükmesi ve çok katmanlı algılayıcı yöntemleri kullanılarak heceyi temel birim olarak alan ayrık Türkçe konuşma tanıma uygulamaları gerçekleştirilmiştir. 200 sözcük içeren sözlük üzerinde yapılan uygulamalar için lpc, parcor, cepstrum ve mfcc öznitelikleri kullanılmıştır. Sözcük ses sinyallerinden hece sınırları tespit edilip hece ses sinyalleri veritabanı oluşturulmuştur. Bu veritabanı kullanılarak bir sözcük ses sinyalinin dinamik zaman bükmesi ve çok katmanlı algılayıcı ile hangi hecelere en çok benzediği tespit edilmiştir. Hecelerin tanınmasından sonra, heceler art arda eklenerek tanınan sözcük elde edilmiştir. Yapılan uygulamalar karşılaştırılmış ve en iyi sonuç tahmin edildiği üzere mfcc özniteliği kullanılarak elde edilmiştir. Dinamik zaman bükmesi ile %95,1 ve çok katmanlı algılayıcı yöntemi ile %92,6 oranında doğru tanıma başarıları sağlanmıştır.

Abstract

We have implemented syllable based isolated word Turkish speech recognition systems with Dynamic Time Warping (DTW) and Multilayer Perceptron (MLP) in this study. Lpc, parcor, cepstrum and mfcc features are used for these applications on the dictionary which includes 200 words. After recording the word utterances, the onsets of syllables are determined and the syllable feature database is constructed. Using this database, the most similar syllables are decided by DTW and MLP. The recognized syllables are concatenated in order. If the constructed word is in the dictionary, the recognized word is found. According to the features, the best results are obtained by DTW with mfcc features. The recognition accuracy rates are 95.1% and 92.6% by DTW and MLP respectively.

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 eğitimi beraberinde getirmektedir.

Türkçe sondan eklemeli diller [1] grubuna girdiğinden geniş ölçekli konuşma tanıma sistemlerinde sözcüğü temel birim olarak almak problem oluşturabilir. Çünkü, bir sözcüğe ekler getirmek suretiyle onlarca yeni sözcük türetilebilir. Bu yüzden çalışmamızda sistemin temel birimi olarak fonem ve sözcük arasında bir birim olan hece seçilmiştir. Günümüze kadar yayınlanmış heceleri temel alan konuşma tanıma [2], [3] ve [4] sistemleri çok fazla değildir. Konuşma tanıma sistemleri konuşmacıya bağımlı veya bağımsız olabilir. Konuşma tanımada yaygın olarak tercih edilen öznitelikler doğrusal öngörülü kodlama (lpc) [4], [5], [6], [7] ve [8], parcor ve cepstrum [4] ve [9], ve mfcc (mel frequency cepstral coefficient) öznitelikleridir. En çok kullanılan yöntemler dinamik zaman bükmesi [10], [11] ve [12], yapay sinir ağları [13] ve [14] ve saklı markov modelidir [1] ve [15].

Bu çalışmada dinamik zaman bükmesi ve çok katmanlı algılayıcı yöntemi kullanılarak ayrık sözcük tanıma uygulaması 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 sözcük kaydedilmiştir ve sistemin en başarılı olduğu mfcc özniteliklerine ve dinamik zaman bükmesi yaklaşımına göre doğru tanıma oranı %95,1 olmuştur. Çok katmanlı algılayıcı yaklaşımı kullanılarak elde edilen tanıma başarıları ise %92,6 olmuştur.

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

2. Sistem Mimarisi

Bu çalışma Şekil 1'de görüldüğü üzere genel anlamda dört aşamadan oluşmaktadır. Birinci aşamada 200 sözcüğe ait hecelerin öznitelik veritabanı oluşturulur. İkinci aşamada her bir heceye ait medyan hece öznitelik vektörleri tespit edilir. Üçüncü aşamada ise konuşma tanıma biriminde DTW veya MLP yöntemlerinden biri kullanılır. Son aşamada tanımayı iyileştirmek için Art-İşleme (postprocessing) uygulanır.

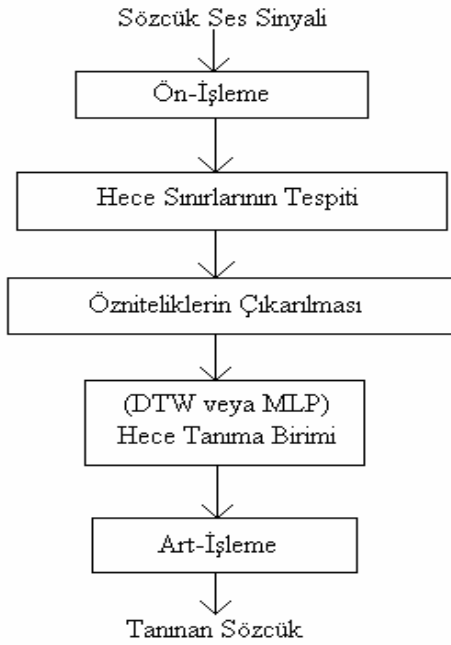
Her bir hecenin özniteliklerinin hesaplanması için sözcükler wave formatında 2 saniyelik zaman aralığında bir konuşmacı tarafından 11025 Hz'de örneklenip 16 bit darbe

kod kiplenimi (PCM) kullanılıp nicemlenerek ve Ön-İşlemeden geçirilmiştir. Ön-İşlemede 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 öznitelik çıkartma işlemi yapılmadan önce ses örneklerine önvurgulama yapılmaktadır. Daha sonra hece sınırları tespit edilir. 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 öznitelik değerlerinden 8'er tane lpc, parcor, cepstrum ve mfcc öznitelikleri hesaplanır.



Şekil 1: Sistemin genel yapısı.

Sistem, dinamik zaman bükmesi yöntemini kullanarak hecelerin en benzer olanlarını hece öznitelik medyan şablonlarına göre elde eder ve sıralar. Tanınacak olan sözcüğün hecelerinin en benzer 5 tanesi tekrar tanıma işleminden geçirilir. Fakat bu sefer beş hece arasında da ayrıntılı tanıma gerçekleştirilir. Yani Hece Ses Öznitelik Şablonu veritabanından tanıma işlemi yapılır. Dolayısıyla bu beş heceden en çok hangisine benzer olduğu tespit edilir. Art-İşleme (Bölüm 5) kullanılarak tanıma işlemi sonlandırılır.

Yapay sinir ağı (YSA) çok katmanlı algılayıcılar (MLP) üzerinde ileri beslemeli geri yayımlı öğrenme algoritmasını çalıştıracak şekilde tasarlandı. Tasarlanan YSA bir giriş katmanı, iki gizli katman ve bir çıkış katmanı olmak üzere 4 katmandan oluşmaktadır. Giriş katmanındaki nöron sayısı 270, çıkış katmanında nöron sayısı bir, gizli katmanlardaki nöron sayıları 20 olarak alınmıştır. Tüm

katmanlarda etkinleştirme fonksiyonu olarak “logsig” kullanılmıştır. Ayrıca nöronların ilk ağırlıkları da rastsal olarak atanmıştır. Her bir hecenin öznitelikleri YSA’ya verilmek suretiyle eğitilmiştir ve tanıma işlemi gerçekleştirilmiştir.

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 hecelerin sınırlarının tespiti işlemidir. Aşağıda her ikisinin algoritması verilmiştir.

Hece sınırlarının tespiti yöntemiyle sözcükteki hece sayıları bakımından elde edilen doğru tanıma yüzdesi yaklaşık olarak %99 olmuştur. Örneğin, beş heceli bir sözcük hece sınırları tespiti algoritması kullanılarak beş heceli olduğu ve bu hecelerin sınırları tespit edilebilmektedir.

3.1. Sözcük Başlangıç ve Bitiş Noktası Tespiti Algoritması

1. x , ses dosyasını temsil eden vektördür. Çalışmamızda $N = 22050$ dir. (2 saniyelik zaman aralığı)

$$x = (x_1, x_2, x_3, \dots, x_N) \quad (2)$$

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

$$\lambda = \left(\sum_{i=1}^{200} x_i \right) / 200 \quad (3)$$

$$\tilde{x} = x - \lambda = (x_1 - \lambda, \dots, x_N - \lambda) = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N) \quad (4)$$

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

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

E_b ve E_s : Sırasıyla başlangıç ve son eşik değerleridir.

$$[M, I] = \max(\tilde{x}) \quad (5)$$

$$E_b = M / I, \quad E_s = M / (N - I) \quad (6)$$

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

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_p) \text{ ve } p = N / 350 \quad (7)$$

$$\bar{x}_i = \left(\sum_{k=i*350}^{(i+1)*350-1} \tilde{x}_k \right) / 350, \quad i=1,2,\dots,p \quad (8)$$

5. $i=1,2,\dots,p-1$ için,

$$\bar{x}_E = (\bar{x}_{E_1}, \bar{x}_{E_2}, \dots, \bar{x}_{E_{p-1}}) \text{ ve } \bar{x}_{E_i} = \bar{x}_{i+1} / \bar{x}_i \quad (9)$$

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 kesişimin olduğu indeksler tespit edilir.

$\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N)$ ses vektörünü kullanarak sıfır kesişi 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 kesişi noktaları arasında olan uzaklıklar hesaplanarak yeni bir sıfır kesişi 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 kesişi eşik değeri, $T=100$ olarak kabul edildi. SB : En son olarak sesin başladığı indeks değeri olacaktır.

$SB = S_b$

For $k = S_b$ DownTo 1

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

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.

$SS = S_s$

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

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

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. $n = (n_1, n_2, \dots, n_k) = (\tilde{x}_{SB}, \tilde{x}_{SB+1}, \dots, \tilde{x}_{SS})$

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

$$\bar{n} = (\bar{n}_1, \bar{n}_2, \dots, \bar{n}_p) \text{ ve } p = k/900 \quad (10)$$

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

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

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

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 bölgeler 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 art arda gelen k 'inci -1 değerlerine sahip indekslerin ortasındaki indeks W ise

$$s_k = SB + 900 * W$$

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 $\tilde{s}_1 = SB$ ve $\tilde{s}_{H+1} = SS$ 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 hecelerin sınır indeksleri \tilde{s} vektörü şeklinde bulunur. k 'inci hecenin hece başlangıcı \tilde{s}_k ve hece bitişi \tilde{s}_{k+1} indeksi olacaktır. Sözcükte H tane de hece vardır.

4. Özniteliklerin Çıkarılması

İlk olarak ses vektörü önvurgulama yöntemiyle filtrelendir. Sonra, 20 ms'lik çerçevelere bölünür. 10 ms'lik örtüşme kullanılır. Her çerçeveye Hamming pencereleme uygulanır. Karşılıklı ilinti [5] ve [6] işlemiyle özilinti vektörü hesaplanır. Levinson metoduyla [6] 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.

5. Art-İşleme

Hecelerin tanıma süreci bittikten sonra heceler art arda eklenerek tanınacak kelime tespit edilmeye çalışılır. Tanınması istenen sözcüğün bütün heceleri için DTW veya MLP ile en çok benzeyen 5 hece tespit edilir. En üst sıradan olmak üzere heceler sıralarına göre eklenir. Hece eklenmesiyle oluşan sözcük sözlüğümüzde var ise tanınan sözcük bu sözcüktür. Aksi halde tanıma süreci gerçekleşmez.

Örnek olarak "kitaplık" sözcüğünü alalım. "ki", "tap" ve "lık" heceleri Tablo 1'de görüldüğü gibi en çok benzeyen heceler bulunmuş olsun. "ki" ve "lık" heceleri ilk sırada bulunmasa bile Art-İşlemeyle "kitaplık" sözcüğü bulunmuş olur. Çünkü "kitaplık" diye bir sözcük sözlüğümüzde bulunmamaktadır.

Tablo 1: Hecelerin tanınma sırası

Hece Sırası	Heceler		
	"ki"	"tap"	"lık"
1.	kı	tap	lik
2.	ki	tap	lak
3.	ki	tep	lık
4.	ki	ta	lik
5.	kı	ta	lık

6. Deney Sonuçları

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 hecelerdir.

Test veritabanındaki sözcüğün hecelerine en küçük uzaklıkta olan heceler birleştirilirse en yakın metin sözcüğü elde edilir. Tablo 2'de kullanılan özniteliklere göre ve art işlem kullanılıp kullanılmadığına bağlı olarak sistemin tanıma oranları verilmiştir. Buna göre Art-İşleme kullanılarak tanıma başarıları yaklaşık %10 artmıştır. En büyük başarı mfcc özniteliği üzerinde Art-İşlem kullanılarak elde edilmiştir ve başarı oranı %95,1 olmuştur.

Tablo 2: Sistemin başarı oranları

Yöntem	Öznitelikler			
	lpc	parcor	cepstrum	mfcc
DTW	%64,9	%75,4	%79,9	%90,3
DTW (Art-İşleme)	%89,1	%89,4	%90,6	%95,1
MLP	%60,4	%70,3	%70,5	%80,2
MLP (Art-İşleme)	%74,3	%88,1	%89,7	%92,6

7. Tartışma ve Sonuçlar

Bu çalışmada, hece tabanlı Türkçe ayrık kişiye bağımlı konuşma tanıma sistemi geliştirilmiştir. Yöntem olarak DTW ve MLP kullanılmıştır. Öznitelik olarak, doğrusal öngörülü kodlama(lpc), parcor, cepstrum ve mfcc öznitelikleri seçilmiştir ve uygulamalar gerçekleştirilip karşılaştırılmıştır. Sisteme dahil edilen art-işleme yöntemi sistemin başarımını oldukça artırmıştır. Sistemin en başarılı özniteliği mfcc olmuştur ve doğru tanıma oranı %95,1 olarak gözlemlenmiştir.

Çalışmanın daha sonraki aşamasında saklı markov modeli yöntem olarak kullanılacaktır. Bütün hecelerinin kullanılıp modellenmesiyle geniş dağarcıklı Türkçe konuşma tanıma sistemi geliştirilmesi mümkün olacaktır.

8. Kaynakça

[1] Mengusoglu, E., and Derro, O., "Turkish LVCSR: Database preparation and Language Modeling for an Agglutinative Language", *ICASSP'2001, Student Forum*, Salt-Lake City, May 2001.

[2] Rosenberg, A. E., Rabiner, L. R., Levinson, S. E. and Wilpon, J. G., "A preliminary study on the use of demisyllables in automatic speech recognition," *Conf. Rec. Int. Conf. on Acous., Speech, and Sig. Processing*, Atlanta, GA, Mar 1981, pp. 967-970.

[3] Jones, R. J., Downey, S. Mason, J. S., "Continuous Speech Recognition using Syllables", *Proc. Eurospeech*, volume 3, pages 1171-1174, 1997.

[4] Svendsen, T., Paliwal, K. K., Harborg E. and Husoy, P. O., "A modified acoustic sub-word unit based speech recognizer", *Proc. IEEE Intern. Conf. on Acoustics, Speech and Signal Processing*, Glasgow, U.K., pp. 108-111, May 1989.

[5] Harma, A., "Linear predictive coding with modified filter structures", *IEEE Trans. Speech Audio Processing*, 9(8):769 --777, November 2001.

[6] Rabiner, L. and Juang B. H., *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[7] Proakis, J. G. And Manolakis, D. G., *Digital Signal Processing: Principles and Application*, Prentice-Hall, Upper Saddle River, NJ, 1996.

[8] Harma, A., "A comparison of warped and conventional linear predictive coding ", *IEEE Trans. Speech and Audio Processing*, July 2001.

[9] Paliwal, K. K., "On the performance of the quefrency-weighted cepstral coefficients in vowel recognition", *Speech Communication*, Vol. 1, No. 2, pp. 151-154, Aug. 1982.

[10] 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.

[11] Alsteris, L. D. and Paliwal, K. K., "Evaluation of the modified group delay feature for isolated word recognition", *Proc. Intern. Symp. on Signal Processing and Its Applications (ISSPA-2005)*, Sydney, Australia, Aug. 2005.

[12] 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, pp. 623-635, Dec 1980.

[13] Mitchell T.M., *Machine Learning*, McGraw-Hill, ISBN: 0070428077, 1997.

[14] Ham F.M., Kostanic I. *Principles of Neurocomputing for science and Engineering*, McGraw-Hill, ISBN: 0070259666, 2001.

[15] Shafran, I., *Clustering wide context and HMM topologies for spontaneous speech recognition*, University of Washington, Ph.D. Thesis, 2001.